UNIVERSITY OF BEIRA INTERIOR

Department of Informatics

Performance Assessment of One-Way Resource Reservation Protocols in IP over Optical Burst Switched Networks

Joel José Puga Coelho Rodrigues

(Master of Science)

Thesis submitted to the University of Beira Interior in candidature for the Degree of Doctor of Philosophy in Informatics Engineering

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

Covilhã, Portugal
2006

Thesis supervised by

Prof. Dr. Mário Marques Freire

Associate Professor of the Department of Informatics of the

University of Beira Interior, Portugal.

Tese realizada sob a orientação do

Professor Doutor Mário Marques Freire

Professor Associado do Departamento de Informática da

Universidade da Beira Interior

To Celina, Catarina, and José.

Ad maiorem Dei gloriam.

Acknowledgements

First of all, I would like to express my heartfelt gratitude to my supervisor Professor Mário Marques Freire for his expertise, guidance, and encouragement. He has always inspired me and provided critical and rapid feedback on my drafts.

I am most grateful to the University of Beira Interior, the Institute of Telecommunications – Covilhã Lab, the Foundation of Science and Technology, the Calouste Goulbenkian Foundation, and the Luso-American Foundation for many kinds of all the support that was given to me. This work was also supported by the *Network of Excellence EuroNGI* — Design and Engineering of the Next Generation Internet —, the Sixth Framework Programme, Priority 2, Information Society Technologies (IST) from the European Union, the *PIONEER Project* — Performance Issues in Optical Networks with Internet Traffic — project of the Institute of Telecommunications, site of Coimbra, and the *CONDENSA Project* — IPv4/IPv6-Compliant Optical Burst Switching Network Design with Enhanced Signaling Architectures — project of the Foundation of Science and Technology (FCT POSC/EEA-CPS/60247/2004).

I owe particular thanks to Dr. George N. Rouskas and Dr. Jing Teng for making available to me their OBS simulator when I was developing our own simulator. I thank my graduate student Nuno Garcia for his cooperation on the development of our simulator.

I thank my colleagues of the Department of Informatics (University of Beira Interior) for their support during this work.

My special thanks to Professors Maria Isabel Figueiredo Silva, Manuel Carlos Lemos, and Isabel Lopes Fael who proofread the drafts of this thesis.

I am grateful to my friends, namely from my Christian Life Community (CLC), for their encouragement and support.

Last, but not least, I am most grateful to my family, especially my wife and my children, for their support, love, and encouragement. I am also grateful for their cooperation and understanding for the amount of hours during which I was absent working on this research. Without that, this work would not be possible.

Abstract

In this thesis, the performance assessment of one-way resource reservation protocols in IP over optical burst switched (OBS) networks is studied. The study is focused on the most important one-way resource reservation protocols (Just-in-Time - JIT, JumpStart, JIT$^+$, Just-Enough-Time (JET), and Horizon), and OBS networks with mesh and ring topologies were analyzed. The mesh topologies considered are the following: chordal rings (with nodal degree three, four, five, and six), mesh-torus (with 16 and 25 nodes), NSFNET (with 14 and 16 nodes), ARPANET, the european optical network, and a topology proposed for the Portuguese *Fundação para a Computação Científica Nacional* (FCCN-NET).

This study takes into account various parameters that influence the performance of OBS networks that are the following: offset length, edge to core node delay, propagation delay between core nodes, number of data channels per link, setup message processing time, optical cross-connect (OXC) configuration time, network size and topology. The work was carried out according to several approaches which consisted of the analysis of the influence of the number of data channels per link, the burst generation ratio (λ/μ), the number of nodes, the nodal degree gain, the setup message processing time, the OXC configuration time, the chord length gain, and the chord length (these latter two may only be applied to chordal ring topologies). This study was carried out by simulation. Due to the inexistence of suitable simulators, at the beginning of this study, an object-oriented model for simulation of IP over OBS networks, called OBSim was proposed, and further implemented and validated.

A new one-way resource reservation protocol, called Enhanced Just-in-Time (E-JIT), was proposed. E-JIT maintains the JIT simplicity in terms of implementation and it reduces the period of time in which the data channel remains in reserved status and optimizes data channel utilization. In this study, E-JIT is shown to perform better than JIT.

Resumo

Nesta tese estuda-se o desempenho de protocolos de reserva unidireccional de recursos em redes IP sobre infra-estruturas com comutação óptica de agregados de pacotes (redes OBS). O trabalho centra-se nos mais importantes protocolos de reserva unidireccional de recursos (*Just-in-Time* - JIT, JumpStart, JIT$^+$, *Just-Enough-Time* – JET, e Horizon), sendo analisadas redes OBS com topologias em malha e em anel. As topologias em malha consideradas são as seguintes: anéis com cordas (de grau três, quatro, cinco e seis), teia toroidal (com 16 e 25 nós), NSFNET (com 14 e 16 nós), ARPANET, rede óptica europeia e uma topologia proposta para a rede de interligação da Fundação para a Computação Científica Nacional.

Este estudo tem em conta diversos parâmetros que condicionam o desempenho das redes OBS e que são os seguintes: o tempo de atraso entre a mensagem de *setup* e o respectivo *burst*, o tempo de atraso entre os nós fronteira e o respectivo nó de interligação, o tempo de propagação ao longo do meio óptico de comunicação entre nós de interligação, o número de canais de dados por ligação, o tempo de processamento da mensagem de *setup*, o tempo de configuração do comutador óptico, a dimensão e a topologia da rede. O trabalho foi efectuado segundo várias vertentes, sendo analisada a influência do número de canais de dados por ligação, do número de saltos (*hops*) para interligar todos os nós da rede, do rácio de geração de *bursts* (λ/μ), do número de nós, do ganho do grau nodal, do tempo de processamento da mensagem de *setup*, do tempo de configuração do comutador óptico, do ganho do comprimento da corda e em função do comprimento da corda (estas duas últimas apenas se aplicam a topologias em anel com cordas). Para efectuar o estudo foi escolhida uma abordagem baseada em simulação. Não havendo simuladores adequados, à data de início do presente estudo, foi proposto e, posteriormente, implementado e validado um modelo orientado a objectos para simulação de redes IP sobre OBS, designado por OBSim.

Foi proposto um novo protocolo de reserva unidireccional de recursos, chamado *enhanced just-in-time* (E-JIT). O E-JIT mantém a simplicidade do JIT em termos de implementação e reduz o período de tempo em que o canal de dados permanece no estado reservado, optimizando a sua utilização. Mostra-se ainda que o desempenho do E-JIT é melhor que o do JIT.

Keywords

Optical Burst Switching, Optical Internet, Resource Reservation Protocols, Performance

Palavras Chave

Comutação de Agregados de Pacotes, Internet Óptica, Protocolos de Reserva de Recursos, Desempenho

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AAP	:	Adaptive-Assembly-Period
ABT	:	ATM Block Transfer
ARDA	:	Advanced Research and Development Agency
ARPA	:	United States Defense Advanced Research Project Agency
ATM	:	Asynchronous Transfer Protocol
BCG	:	Data Burst Channel Group
BTCP	:	Burst Transfer Control Protocol
CCG	:	Control Channel Group
CORD	:	Contention resolution by delay lines
CRC	:	Cyclic Redundancy Code
DiffServ	:	Differentiated Services
DnT	:	Degree n topology
DQDB	:	Distributed Queue Dual Bus
DVMRP	:	Distance vector multicast routing protocol
EFSM	:	Extended finite state machine
E-JIT	:	Enhanced JIT
EON	:	European Optical Network
ESCON	:	Enterprise Serial Connection
FAP	:	Fixed-Assembly-Period
FCCN	:	Portuguese *Fundação para a Computação Científica Nacional*
FCFS	:	First-come, first-served
FDDI	:	Fiber Distributed Data Interface
FDL	:	Fiber Delay Lines
FEC	:	Forwarding equivalent classes
FFUC	:	First-fit unscheduled channel
FPQ	:	Fair Packet Queuing

GMPLS	:	Generalized MultiProtocol Label Switching
HBP	:	Hybrid Burst Priority
HIPPI	:	High Performance Parallel Interface
HTTP	:	Hypertext Transfer Protocol
IETF	:	Internet Engineering Task Force
IntServ	:	Integrated Services
IP	:	Internet Protocol
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
JITPAC	:	Just-in-Time Protocol Acceleration Circuit
LAN	:	Local Area Network
LAUC	:	Latest available unscheduled channel
LAUC-VF	:	Latest available unscheduled channel with void filling
LOBS	:	Labeled Optical Burst Switching
MAN	:	Metropolitan Area Network
MAP	:	Maximum Assembly Period
Max-EV	:	Maximum ending void
Max-SV	:	Maximum starting void
MBL	:	Minimum Burst Length
MBMAP	:	Min-BurstLength-Max-Assembly-Period
MEMS	:	Micro-Electro-Mechanical Systems
Min-EV	:	Minimum ending void
Min-SV	:	Minimum starting void
MONET	:	Multiwavelength Optical Networking
MPλS	:	Multiprotocol Lambda Switching
MPLS	:	MultiProtocol Label Switching
MSC	:	Multicast Sharing Classes
M-UCAST	:	Multiple Unicasting
NAK	:	Negative acknowledge
NCSU	:	North Carolina State University
ns-2	:	Network simulator version 2
NSF	:	USA National Science Foundation

OBS	:	Optical Burst Switching
OCBS	:	Optical Composite Burst Switching
OCS	:	Optical Circuit Switching
OOP	:	Object-Oriented Programming
OPS	:	Optical Packet Switching
OSI	:	Open Systems Interconnection
OXC	:	Optical Cross Connect
pJET	:	prioritized JET
QoS	:	Quality of Service
RAM	:	Random Access Memory
SAN	:	Storage Area Network
SCU	:	Switch Control Unit
SDH	:	Synchronous Digital Hierarchy
SLOB	:	Switch with large optical buffers
S-MCAST	:	Separate Multicasting
SMTP	:	Simple Mail Transfer Protocol
SONET	:	Synchronous Optical Network
SS/TDMA	:	Satellite-Switched Time Division Multiple
TAG	:	Tell And Go
TAW	:	Tell And Wait
TCP	:	Transmission Control Protocol
TDM	:	Time-Division Multiplexing
ToS	:	Type of Service
TS-MCAST	:	Tree-Shared Multicasting
UML	:	Unified Modeling Language
WAN	:	Wide Area Network
WDM	:	Wavelength Division Multiplexing
WR-OBS	:	Wavelength-Routed Optical Burst Switching

Extended Abstract in Portuguese

O presente resumo alargado em língua portuguesa sintetiza a tese de Doutoramento intitulada "Avaliação do Desempenho de Protocolos de Reserva Unidireccional de Recursos em Redes IP sobre Infra-estruturas com Comutação Óptica de Agregados de Pacotes" (*Performance Assessment of One-Way Resource Reservation Protocols in IP over Optical Burst Switched Networks*).

Começa-se por enumerar os objectivos e referir as principais contribuições desta investigação. Em seguida, faz-se o enquadramento do tema da tese e descrevem-se os principais aspectos das redes baseadas no paradigma de comutação de agregados de pacotes, nomeadamente, a arquitectura da rede, o processo de criação dos agregados de pacotes, os protocolos de reserva de recursos e a resolução do problema da contenção. Neste ponto, é também descrito um novo protocolo de reserva de recursos proposto nesta tese, o qual é designado por *Enhanced Just-in-Time* (E-JIT). Posteriormente, é apresentada uma breve descrição do simulador desenvolvido e que foi utilizado como ferramenta de apoio ao estudo do desempenho dos protocolos de reserva de recursos estudados nesta tese. Depois, apresenta-se uma avaliação de desempenho dos protocolos de reserva unidireccional de recursos, propostos na literatura, em redes com comutação óptica de agregados de pacotes com topologias regulares e em malha, bem como a avaliação do desempenho do protocolo E-JIT. Finalmente, são sintetizadas as principais conclusões e apontadas direcções para trabalho futuro.

Definição do Problema e Objectivos

No início deste programa de investigação, em Setembro de 2002, já havia sido prestada uma atenção considerável às redes com comutação óptica de agregados de

pacotes (*optical burst switching* - OBS). No entanto, muitos dos estudos publicados eram de natureza teórica [1-6] e, tipicamente, consideravam apenas redes OBS muito simples, analisando geralmente um porto de saída de um nó OBS. Deste modo, determinados parâmetros de rede, tais como o tamanho da rede e a sua topologia, o número de canais por ligação, etc, foram geralmente ignorados e foi necessário procurar novas metodologias para avaliar o desempenho deste tipo de redes. Além disso, muitos dos estudos, publicados até essa altura, ignoravam alguns parâmetros importantes como são o caso do tempo de atraso entre a mensagem de setup e o respectivo *burst* (*offset time*), o tempo de atraso entre os nós fronteira e o respectivo nó de interligação, o tempo de propagação ao longo do meio óptico de comunicação entre nós de interligação, o número de canais de dados por ligação, o tempo de processamento da mensagem de *setup*, o tempo de configuração do comutador óptico, a dimensão e a topologia da rede, que podem causar um impacto significativo no desempenho da rede. Tendo em conta estas circunstâncias, estudos anteriores mostraram que o protocolo *Just-in-Time* (JIT) tinha um desempenho pior, em termos de probabilidade de perda de *bursts*, do que o *Just-Enough-Time* (JET) ou o Horizon. De facto, o escalonamento avançado e o preenchimento de vazios usado no JET e no Horizon podem conduzir, à primeira vista, à ideia que o JET e o Horizon teriam um melhor desempenho que o JIT. Contudo, é razoável assumir que o processamento das mensagens de *setup* no JET e no Horizon será mais longo que no JIT, dadas as operações complexas e/ou grande número de acessos à memória e, por consequência, não é claro que os algoritmos mais eficientes possam compensar o processamento adicional necessário.

Por outro lado, se o tempo de configuração do comutador óptico é maior que o tamanho médio do *burst*, o impacto da eficiência do escalonamento na probabilidade de perda de *bursts* pode ser pequeno. Além disso, os efeitos subsequentes, devido ao facto de haver diversos comutadores ópticos no caminho óptico, necessitam de ser investigados, uma vez que cada comutador óptico necessita de ser configurado pela mensagem de *setup* antes de ser manipulado o *burst*. Adicionalmente, tem-se verificado um aumento do interesse pelas topologias em malha para redes ópticas de área alargada (*wide area networks* - WANs) [7-16]. Por consequência, há a necessidade de efectuar estudos de desempenho mais detalhados, tendo em conta o tempo de separação entre a mensagem de *setup* e o respectivo *burst*, o tempo de processamento das mensagens de *setup*, o tempo de configuração do comutador óptico e os efeitos em cascata dos comutadores ópticos em redes OBS com topologias em malha ou em anel.

O principal objectivo desta tese é apresentar um estudo do desempenho de redes OBS com topologias em malha e em anel para os mais importantes protocolos de reserva unidireccional de recursos (JIT, JumpStart, JIT$^+$, JET e Horizon), tendo em conta o tempo de atraso entre a mensagem de setup e o respectivo *burst*, o tempo de atraso entre os nós fronteira e o respectivo nó de interligação, o tempo de propagação ao longo do meio óptico de comunicação entre nós de interligação, o número de canais de dados por ligação, o tempo de processamento da mensagem de *setup*, o tempo de configuração do comutador óptico, a dimensão e a topologia da rede.

Para atingir este objectivo, foram identificados e executados os seguintes objectivos intermédios:

- Análise detalhada da arquitectura de uma rede OBS;
- Análise detalhada dos cinco protocolos de reserva unidireccional de recursos mais importantes;
- Proposta, implementação e validação de um modelo orientado a objectos para simulação de redes IP sobre OBS usando uma topologia regular ou irregular qualquer, para os cinco protocolos de reserva unidireccional de recursos, tendo em conta os parâmetros OBS ignorados anteriormente como são o caso do tempo de separação entre a mensagem de *setup* e o respectivo *burst*, o tempo de processamento das mensagens de *setup* e o tempo de configuração do comutador óptico;
- Estudo do desempenho dos cinco protocolos de reserva unidireccional de recursos em redes IP sobre OBS com topologias em anel e anel com cordas;
- Estudo do desempenho dos cinco protocolos de reserva unidireccional de recursos em redes IP sobre OBS com topologias em malha.

Apesar de não ser um objectivo inicial, depois da análise dos cinco protocolos de reserva unidireccional de recursos acima mencionados, foi identificada a possibilidade de optimização na operação do JIT, o que conduziu à proposta de um novo protocolo de reserva unidireccional de recursos chamado *Enhanced Just-in-Time* (E-JIT). O desempenho deste novo protocolo foi avaliado e comparado com o JIT, utilizando a ferramenta de simulação desenvolvida no âmbito deste trabalho. Com este estudo, concluiu-se que o desempenho do E-JIT é melhor que o do JIT, confirmando a importância que tem a utilização do canal de dados no estado livre,

durante o tempo de processamento da mensagem de *setup*, tem no desempenho destes protocolos.

Principais contribuições

Esta secção é dedicada às principais contribuições científicas da presente tese. Neste sentido, os próximos parágrafos descrevem, na opinião do Autor, as principais contribuições para o avanço do estado da arte na área da Internet óptica.

A primeira contribuição é uma análise detalhada e compreensiva das redes com comutação óptica de agregados de pacotes, que são apresentadas no Capítulo 2. Uma versão curta desta análise foi publicada como entrada na *Encyclopedia of Multimedia Technology and Networking* pela *Idea Group Reference* [17].

A segunda contribuição desta tese é a proposta, implementação e validação de um modelo orientado a objectos para simulação de IP sobre redes OBS em malha, cuja descrição é apresentada no Capítulo 3. Este modelo foi apresentado no *10th IEEE Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD'2004),* que fez parte da *IEEE Global Telecommunications Conference (GLOBECOM'2004)* [18]. Uma versão alargada deste artigo foi aceite para publicação como capítulo do livro *Simulation and Modeling: Current Technologies and Applications* [19], publicado pela *Idea Group, Inc*.

A terceira contribuição consiste na proposta de duas novas métricas para avaliação do desempenho de redes OBS. Estas medidas de desempenho, o ganho do grau nodal e o ganho do comprimento da corda (que apenas se aplica a topologias em anel com cordas) são descritos na Secção 2 do Capítulo 4. O ganho do comprimento da corda foi igualmente apresentado na revista *Kluwer Telecommunications System Journal* [20]. Em relação ao ganho do grau nodal, um caso especial válido apenas para redes OBS com topologias em anel com cordas de grau três foi relatado em [20], sendo posteriormente estendido para anéis com cordas de grau quatro em [21-24]. A generalização do ganho do grau nodal válido para redes OBS com topologias em malha foi apresentada em [25, 26].

A quarta contribuição é um estudo do desempenho de redes em anel e anel com cordas de grau três para os protocolos JIT, JumpStart, JIT$^+$, JET e Horizon. Este estudo é mostrado na Secção 3 do Capítulo 4. Um primeiro estudo para avaliação do desempenho, considerando os protocolos JIT e JET, foi apresentado na *3rd International Conference on Networking* [21]. Este artigo foi considerado um dos

melhores entre os 169 aceites a partir de 320 submissões e foi seleccionado para publicação, numa versão estendida, na revista *Kluwer Telecommunications System Journal* [20]. Esta versão estendida teve em conta os cinco protocolos de reserva unidireccional de recursos: JIT, JumpStart, JIT$^+$, JET e Horizon.

A quinta contribuição é um estudo do desempenho de redes OBS em anel com cordas de grau quatro para os cinco protocolos de reserva unidireccional de recursos em análise. Este estudo, exposto na Secção 4 do Capítulo 4, foi apresentado na *12th International Conference on Telecommunications* [22].

A sexta contribuição compreende a avaliação do desempenho dos cinco protocolos de reserva unidireccional de recursos considerados, em redes com comutação óptica de agregados de pacotes com topologias em malha com grau nodal até quatro. Uma versão preliminar deste trabalho, considerando apenas topologias em malha com grau nodal até três, foi apresentado na *International Conference on Information Networking (ICOIN 2004)* [23]. Este artigo foi posteriormente seleccionado para publicação, após revisão, no livro *Information Networking: Networking Technologies for Broadband and Mobile Networks* incluído na série *Lecture Notes in Computer Science* da Springer-Verlag [24]. Um estudo adicional, para topologias com grau nodal até quatro, foi apresentado na *7th International Conference on High-Speed Networks and Multimedia Communications* [27].

A sétima contribuição é a proposta de uma topologia de rede OBS para o *backbone* da rede da Fundação para a Computação Científica Nacional (FCCN) e a sua avaliação de desempenho. Este estudo (em língua portuguesa) foi apresentado na *7ª Conferência sobre Redes de Computadores (CRC'2004)* [28]. Um estudo adicional, comparando esta proposta com a topologia em anel com cordas de grau três com o mesmo número de nós, foi apresentado na *5th Conference on Telecommunications* [25].

A oitava contribuição é um estudo detalhado do impacto do grau nodal no desempenho de redes com comutação óptica de agregados de pacotes. Versões preliminares deste estudo foram apresentadas na *International Conference on Information Networking* [29] e na *4th International Conference on Networking* [30], para redes com 16 e 20 nós, respectivamente. O primeiro fez parte dos 22% de artigos aceites entre 427 submissões e foi publicado no livro da série *Lecture Notes in Computer Science* da Springer-Verlag. O último foi considerado um dos melhores artigos dos 238 aceites entre 651 submissões e foi seleccionado para publicação, como artigo longo, na revista *Kluwer Telecommunications System Journal* [31]. Continuando este estudo, a influência do grau nodal no ganho do grau nodal (considerando topologias em malha com grau nodal entre três e seis) foi apresentada

na *International Conference on Systems Communications 2005* [26], a qual seleccionou 80 artigos entre mais de 200 submetidos.

A nona contribuição é composta por um estudo que analisa a influência dos tempos de processamento da mensagem de *setup* e de configuração do comutador óptico no desempenho de redes com comutação óptica de agregados de pacotes. Uma versão preliminar deste estudo foi apresentada na *20th International Symposium on Computer and Information Sciences* [32]. Esta conferência aceitou aproximadamente 30% das 491 submissões, sendo este artigo um dos 20% aceites para publicação na série *Lecture Notes in Computer Science* da Springer-Verlag, enquanto os restantes 10% foram seleccionados para publicação numa edição especial da *Advances in Computer Science and Engineering Series* da *Imperial College Press*.

A décima contribuição é a especificação formal e avaliação do desempenho do novo protocolo de reserva de recursos unidireccionais, chamado *Enhanced Just-in-Time* (E-JIT). Esta contribuição foi publicada como relatório técnico [33] e será adicionalmente submetida para publicação numa revista internacional com *refereeing*.

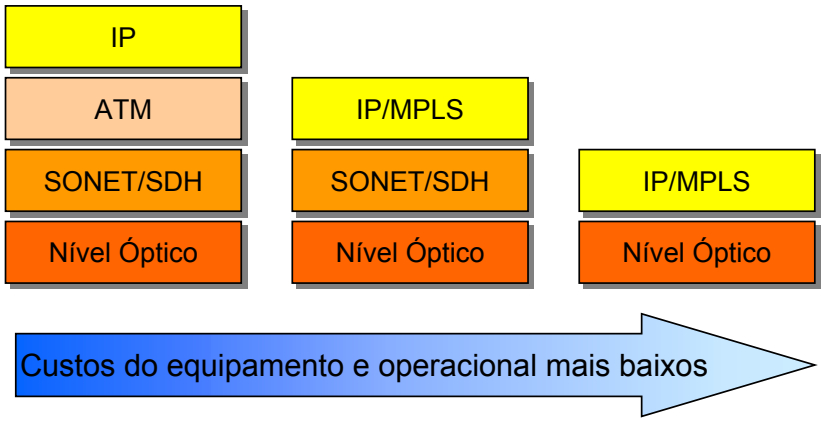
Enquadramento da Tese

Uma rede em fibra óptica, também designada por rede óptica, é formada por um conjunto de nós interligados por fibra óptica para transportar dados sobre estas mesmas fibras. De acordo com a sua evolução, as redes em fibra óptica podem ser classificadas como primeira geração e segunda geração de redes [34, 35]. A primeira geração de redes ópticas tem como principal característica o enfoque dado à resolução do problema das ligações entre nós. Nesta geração, a fibra óptica é usada como meio de transmissão, enquanto que as funções de comutação, processamento e encaminhamento são realizadas ao nível electrónico. Actualmente, estas redes são universalmente usadas em todo o tipo de redes de telecomunicações, com a possível excepção da rede de acesso. Exemplos desta primeira geração de redes em fibra óptica incluem as redes *i*) a SONET (*Synchronous Optical Network*) e o SDH (*Synchronous Digital Hierarchy*), que constituem a infra-estrutura de telecomunicações base na América do Norte, Europa e Ásia; *ii*) a variedade de fibra óptica utilizada em redes *Ethernet*, universalmente usada em redes de área local (*local area networks* - LANs); *iii*) o anel FDDI (*Fiber Distributed Data Interface*) utilizado em LANs e redes de área metropolitana (*metropolitan area networks* -

MANs); *iv*) a tecnologia DQDB (*Distributed Queue Dual Bus*) usada em MANs; e *v*) uma variedade de redes conhecidas como *storage area networks* (SANs) nas quais se destacam a ESCON (*Enterprise Serial Connection*) e a HIPPI (*High Performance Parallel Interface*).

Recentemente, foi reconhecido que as redes ópticas são capazes de fornecer mais funções do que apenas a transmissão ponto-a-ponto. Deste modo, a segunda geração de redes ópticas focaliza-se na resolução dos problemas de encaminhamento [34]. Na primeira geração, determinado nó electrónico não só devia gerir os dados destinados a esse mesmo nó como também os dados destinados a outros nós. Se os dados destinados a outros nós da rede puderem ser encaminhados através do domínio óptico, a carga na parte electrónica desse nó será significativamente reduzida. Este é um dos aspectos chave que conduziu ao desenvolvimento da segunda geração de redes ópticas. Este tipo de redes, usando multiplexagem por divisão no comprimento de onda (*wavelength division multiplexing* - WDM) e encaminhamento por comprimento de onda, está actualmente a ser desenvolvido.

A terceira geração de redes ópticas, também conhecida como a próxima geração da Internet óptica, está a emergir [36-38]. Nesta nova geração, a interacção entre o nível WDM e o nível do protocolo Internet (Internet *protocol* - IP) é do maior interesse, uma vez que leva tanto a baixos custos de gestão como a baixa complexidade. Isto consiste no uso de uma arquitectura com dois níveis, na qual o tráfego IP[1] é transportado directamente sobre redes ópticas, como se mostra na Figura I.



IP: *Internet Protocol*; ATM: *Asynchronous Transfer Mode*; SONET: *Synchronous Optical NETwork*; SDH: *Synchronous Digital Hierarchy*; MPLS: *MultiProtocol Label Switching*

Fig. I. Evolução para uma arquitectura com dois níveis.

[1] Trata-se de um abuso de linguagem usado na literatura da especialidade: de facto o que é transportado não é o IP mas sim o tráfego gerado pelo IP.

Como referido anteriormente, hoje em dia, os dados são transmitidos opticamente em redes de área alargada (WANs), MANs e ainda algumas LANs. Em termos de tecnologias de comutação para redes ópticas existem dois tipos principais: comutação electrónica e comutação óptica. Resumidamente, a comutação electrónica usa uma matriz de comutação digital (ou electrónica) e converte os dados, de sinal óptico para sinal electrónico para comutação, e depois, converte o sinal electrónico em óptico para transmissão. Por outro lado, a comutação óptica usa uma matriz de comutação óptica e os dados são mantidos no domínio óptico. Neste caso, apenas é considerada a comutação óptica.

Com o objectivo de usar a fibra óptica de uma forma mais flexível e permitir a utilização da sua enorme capacidade potencial, alguns paradigmas de comutação óptica têm sido propostos, nomeadamente, a comutação óptica de circuitos (*optical circuit switching*) [39-43], a comutação óptica de pacotes (*optical packet switching*) [37, 43-53] e a comutação óptica de agregados de pacotes (*optical burst switching*) [5, 37, 43, 47, 52, 54-60].

Na comutação óptica de circuitos [39-43] (através do encaminhamento por comprimento de onda em redes com multiplexagem por divisão no comprimento de onda - WDM) um caminho óptico (também referido como canal λ) estabelecido entre os nós origem e destino utiliza um comprimento de onda dedicado para cada ligação ao longo do caminho físico estabelecido. Se estiverem presentes conversores de comprimento de onda em alguns nós da rede, um caminho óptico pode ser constituído por diferentes comprimentos de onda ao longo desse caminho. Neste paradigma de comutação, há três fases distintas: estabelecimento do circuito, transferência de dados e libertação do circuito [43]. Na primeira fase, apenas são trocadas mensagens de controlo para estabelecimento de uma ligação ponto-a-ponto entre o nó origem e o nó destino da ligação (nós de ingresso e egresso, respectivamente), como por exemplo, informação sobre o pedido da ligação e a confirmação da ligação. Este circuito virtual usa um canal dedicado com uma largura de banda fixa (por exemplo, um *time-slot* ou uma frequência) entre a origem e o destino. A transferência de dados caracteriza a segunda fase. Quando a transferência de dados está completa, o circuito é libertado (terceira fase). Para os serviços da Internet, usando comutação óptica de circuitos, a largura de banda seria utilizada de forma ineficiente porque este tipo de aplicações têm uma transmissão de dados de curta duração quando comparada com o tempo necessário para o estabelecimento do circuito [57].

Na comutação óptica de pacotes [37, 43-53, 61] os dados são enviados sem o estabelecimento prévio de um circuito e os dados são transmitidos em pacotes ópticos. Cada pacote contém um cabeçalho com alguma informação de controlo (controlo *in-band*) e é enviado através dos nós intermédios usando a técnica de armazenamento e envio de pacotes (*store-and-forward*). Nesta técnica, quando um pacote chega a um nó, primeiro é armazenado e depois do seu cabeçalho ser processado é enviado para o próximo nó. Isto implica que o comutador apenas é configurado quando o pacote chega. Os pacotes podem ter tamanho fixo (ex.: célula ATM – *Asynchronous Transfer Mode*) ou variável (ex.: pacote IP) com um tamanho máximo limitado. Neste paradigma de comutação não são necessárias as conversões de sinal óptico para electrónico ou electrónico para óptico.

Contudo, a tecnologia actual para comutação óptica de pacotes é imatura para oferecer uma boa solução. Isto deve-se, principalmente, aos três problemas que a seguir se enumeram [46, 47, 52, 62]:

- Sincronização — os comutadores ópticos de pacotes trabalham em modo síncrono. Por exemplo, se os pacotes chegam a diferentes portos de entrada é necessário alinhá-los antes de entrarem na matriz de comutação. No entanto, é tão difícil quanto dispendioso desenvolver um módulo de sincronização [46].
- *Buffer* óptico — na comutação óptica de pacotes é necessário um módulo para *store-and-forward*. Esta técnica é necessária devido aos problemas de contenção nos portos de saída. Contudo, como as actuais memórias de acesso aleatório (*random access memory* - RAM) ainda estão num estado imaturo de desenvolvimento, os *buffers* ópticos são conseguidos através de linhas de atraso baseadas em fibra óptica (designadas na literatura por *fiber delay lines* - FDLs) que permitem um determinado atraso do sinal. O valor do atraso é determinado em função do comprimento da fibra [47, 52].
- Tempo para configurar a matriz de comutação óptica — dado o actual estado da arte em termos de comutadores ópticos de pacotes, é necessário um valor muito optimista de 1ms para definir a ligação entre um porto de entrada e um porto de saída. Por exemplo, segundo Xu [62], a uma velocidade de transferência de dados de 2.5 Gbps, demora à volta de 5μs para transmitir um pacote com 1500 bytes. Assim, usando um comutador de pacotes, menos de 0.5% do tempo é

usado para comutar dados ao passo que o restante é dispendido para configurar a matriz de comutação.

Comutação óptica de agregados de pacotes, aqui também designadoa por OBS [5, 37, 43, 47, 52, 54-60, 63-65], é um compromisso técnico entre a comutação de comprimentos de onda (i.e. comutação de circuitos) e a comutação óptica de pacotes, uma vez que não precisa de *buffers* ópticos ou processamento ao nível do pacote de dados como na comutação óptica de pacotes, e é mais eficiente do que a comutação de circuitos quando o volume de tráfico não exige um canal de dados completo. Em redes OBS, os pacotes (datagramas) IP, são agregados em pacotes muito grandes chamados *bursts* de dados (*data bursts*). Estes *bursts* são transmitidos depois de um pacote com um cabeçalho do *burst*, com um atraso entre eles de algum tempo de separação (*offset time*). Os sinais de controlo e de dados são enviados separadamente em diferentes canais ou comprimentos de onda (Lambdas – λ's)[2]. Cada pacote de cabeçalho do *burst* contém informação sobre o caminho e o agendamento do *burst*, e é processado electronicamente antes da chegada do *burst* correspondente. Actualmente, o OBS ainda não está normalizado [66]. Contudo, existe um conjunto de características universalmente aceites que o distinguem e que são as seguintes [4]:

- Granularidade — o tamanho da unidade de transmissão em OBS (o *burst*) está entre a comutação de circuitos e a comutação de pacotes;
- Separação entre o controlo e os dados — a informação de controlo (cabeçalho) e os dados são transmitidos em canais separados com algum tempo de intervalo;
- Reserva de recursos — os recursos são atribuídos usando um esquema de reserva unidireccional;
- Tamanho variável do *burst* — o tamanho do *burst* não é fixo;
- Não há *buffers* ópticos — os dados enviados não necessitam de nenhum armazenamento temporário nos nós intermédios.

No ponto intitulado "Redes com Comutação de Agregados de Pacotes (OBS)", do presente resumo, serão apresentados os aspectos mais relevantes do OBS, com base na literatura científica publicada sobre o tema.

[2] No que ao OBS diz respeito, os termos canal, comprimento de onda (ou λ) são usados de forma indiferenciada.

Dado o crescimento exponencial do tráfego Internet, a comutação óptica de pacotes está também a ganhar muito interesse [67-69]. Contudo, esta tecnologia apresenta ainda um estado de desenvolvimento imaturo, principalmente dadas as limitações da memória óptica de acesso aleatório (RAM) e os problemas de sincronização mencionados anteriormente [47, 52]. Não obstante, diversas abordagens têm sido propostas para resolver estes problemas, nomeadamente uma, conduzida pela *Internet Engineering Task Force* (IETF) [70], i.e. comutação multiprotocolo com etiquetagem generalizada (dos datagramas) (*Generalized MultiProtocol Label Switching* - GMPLS) [71, 72], e a comutação óptica de agregados de pacotes (*Optical Burst Switching* - OBS) [1, 57, 58, 63, 64, 73-75].

O GMPLS estende a arquitectura de comutação com etiquetagem proposta na comutação multiprotocolo com etiquetagem (*MultiProtocol Label Switching* - MPLS) com o objectivo de incluir outros tipos de redes que não sejam de pacotes, como as redes com encaminhamento por comprimento de onda e as SONET/SDH. O GMPLS define quatro tipos de interface para as seguintes tecnologias [37]: *i*) comutação de pacotes, *ii*) multiplexagem por divisão temporal (*time division multiplexing*), *iii*) comutação de circuitos, e *iv*) comutação de fibras. Estas interfaces encaminham dados baseadas no seu conteúdo, respectivamente *i*) cabeçalho de um pacote/célula, *ii*) *time-slot* de dados, *iii*) comprimento de onda, e *iv*) fibra óptica [37]. O GMPLS estende o plano de controlo do MPLS para suportar cada uma destas quatro interfaces. A extensão do MPLS para as redes ópticas ajuda a minimizar o custo de transição da tecnologia de encaminhamento por comprimento de onda para as tecnologias de comutação de pacotes ou comutação de agregados de pacotes.

O GMPLS permite integrar directamente o nível IP com o nível óptico. Além disso, ele pode ser aplicado a redes com encaminhamento por comprimento de onda. Uma rede com multiplexagem por divisão no comprimento de onda (WDM) que usa MPLS pode ser classificada como [76]: comutação óptica de circuitos com etiquetagem (*labeled optical circuit switching*), comutação óptica de agregados de pacotes com etiquetagem (*labeled optical burst switching*) e comutação óptica de pacotes com etiquetagem (*labeled optical packet switching*), dependendo da tecnologia de comutação utilizada. A comutação óptica de circuitos com etiquetagem é também referida como etiquetagem *lambda* ou comutação multiprotocolo *lambda* (*multiprotocol lambda switching* - MPλS) [77-79].

Esta tese centra-se nas redes com comutação óptica de agregados de pacotes. Como foi descrito anteriormente, neste tipo de redes, a transmissão de um *burst* de dados (num determinado canal de dados) ocorre após a transmissão de uma

mensagem de *setup* (no canal de controlo). Esta mensagem de *setup* contém a informação de encaminhamento e escalonamento para ser processada em cada comutador intermédio no sentido de configurar a matriz de comutação para encaminhar o correspondente *burst* de dados para o porto de saída adequado. Diversos protocolos de reserva unidireccional de recursos foram propostos para redes OBS. Dois tipos de esquemas de reserva podem ser considerados: reserva imediata e reserva atrasada. Os protocolos *just-in-time* (JIT) [74, 75] e JIT+ [80] são exemplos de protocolos de reserva imediata, enquanto que o *just-enough-time* (JET) [57, 73, 81] e o Horizon [1, 54] são exemplos de protocolos de reserva com atraso. O JumpStart [75, 82-86] é um exemplo de um protocolo, proposto a partir do JIT, que pode ser configurado tanto para reserva imediata como para reserva com atraso. Ao longo desta tese, vai ser dada particular atenção ao desempenho destes protocolos de reserva unidireccional de recursos em redes OBS com topologias em malha e em anel.

Redes com Comutação de Agregados de Pacotes (OBS)

Esta secção descreve os principais aspectos das redes baseadas no paradigma de comutação óptica de agregados de pacotes, incluindo a arquitectura da rede, o processo de criação dos agregados de pacotes, os protocolos de reserva de recursos e a resolução do problema da contenção. Também se apresenta um novo protocolo de reserva de recursos proposto nesta tese, designado por *Enhanced Just-in-Time* (E-JIT).

Arquitectura da Rede OBS

Uma arquitectura de comunicação define a estrutura e o comportamento de um sistema real que é visível para outros sistemas ligados em rede, enquanto eles estão envolvidos no processamento e transferência de conjuntos de informação [87]. A arquitectura das redes OBS segue o modelo de referência OSI (*Open Systems Interconnection*) da *International Organization for Standardization* (ISO) [88] e a arquitectura TCP/IP (*Transmission Control Protocol/Internet Protocol*) [87, 89].

Uma rede OBS é considerada uma rede totalmente óptica, na qual os nós de interligação transportam os dados de/para os nós fronteira (que geralmente são

encaminhadores IP ou *routers* IP) interligados por ligações bidireccionais. Uma comunicação entre um nó fronteira *A* e um nó fronteira de destino *B* processa-se sequencialmente do seguinte modo: 1) no nó *A*, os pacotes de entrada são agregados em *bursts* e armazenados no *buffer* de saída; 2) é criado o pacote de controlo e este é enviado de imediato em direcção ao nó destino para estabelecer a ligação para o respectivo *burst*; 3) depois de um determinado tempo de atraso (T_{offset}[3]), o *burst* é transmitido opticamente através dos nós OBS de interligação sem qualquer armazenamento nos nós intermédios, até ao nó fronteira de destino; 4) após a recepção do *burst*, o nó de destino procede à sua desagregação em pacotes e entrega-os aos níveis superiores; 5) os pacotes são enviados electronicamente para os utilizadores destino [60, 90-93].

O nó fronteira OBS funciona com a interface entre um *router* IP típico e o *backbone* OBS [91, 94]. Em [91] são resumidas as seguintes operações que devem ser realizadas por um nó fronteira OBS:

- Agregar os pacotes IP em *bursts* de dados, baseada em alguma política de agregação;
- Gerar e escalonar o pacote de controlo para cada *burst*;
- Converter o tráfego de sinal electrónico em sinal óptico e multiplexá-lo para dentro de um comprimento de onda WDM (*Wavelength Division Multiplexing* – multiplexagem por divisão no comprimento de onda);
- Desmultiplexar os comprimentos de onda dos canais de entrada e converter o sinal do tráfego recebido, de óptico para electrónico;
- Desagregar os *bursts* e enviar os pacotes IP para os *routers* a ele ligados.

Um nó fronteira é constituído por três módulos [90, 93]: módulo de *routing*, módulo de agregação de *bursts* e módulo de escalonamento. O módulo de *routing* selecciona o porto de saída adequado para cada pacote e envia-o para o correspondente módulo de agregação de *bursts*. Este agrega os *bursts* que contêm pacotes que são endereçados para o mesmo nó fronteira de destino. Neste módulo existem diferentes filas de espera de acordo com a classe de tráfego (definida de acordo com a sua prioridade). O módulo de escalonamento constrói o *burst* e o

[3] O T_{offset} representa o tempo necessário para cada protocolo efectuar a reserva de recursos até ao nó fronteira destino. O seu valor depende do protocolo e do número de nós da ligação. O maior interesse é fazer com que o seu valor seja tal que, quando o primeiro bit do *burst* chegar ao nó fronteira de destino, o nó deva estar configurado e pronto para receber esse *burst*.

respectivo pacote de controlo baseado na política de agregação utilizada e envia-o para o porto de saída.

O nó de interligação é formado por dois componentes principais [90, 93, 95]: um *cross-connect* óptico (*optical cross connect* - OXC) e uma unidade de controlo do comutador (*Switch Control Unit* – SCU) ou motor de sinalização (*Signaling Engine*). A SCU implementa o protocolo de sinalização, cria e mantém a tabela de encaminhamentos e configura o OXC. O OXC manipula a matriz óptica de comutação para conduzir os *bursts* de dados para o próximo nó da rede. Algumas matrizes de comutação óptica foram desenvolvidas, nomeadamente as de *broadcast-and-select*, propostas em [96] e as bem conhecidas matrizes de comutação apresentadas em [97-99]. Nesta tese apenas são considerados OXC com matrizes de comutação por divisão espacial não bloqueante (*non-blocking space-division switch matrix*) sem *buffers*. Num nó de interligação, os canais de dados estão ligados à matriz óptica de comutação e os canais de controlo estão ligados à SCU. As funções de uma SCU são semelhantes às de um *router* electrónico convencional [95].

Em [100] são resumidas as principais funções de um nó de interligação e que são as seguintes:

- Desmultiplexar o comprimento de onda dos canais de dados;
- Ligação terminal dos canais de dados e converter o comprimento de onda para passar através da matriz óptica de comutação;
- Ligação terminal dos canais de controlo e converter o sinal da informação de controlo do domínio óptico para o electrónico;
- Escalonar os *bursts* de entrada, enviando instruções para a matriz óptica de comutação e comutar dos canais de *bursts* de dados através dessa matriz óptica de comutação;
- Recriar novos pacotes de controlo para os *bursts* de saída;
- Multiplexar os pacotes de controlo e de *bursts* de saída numa única ou múltiplas fibras.

O nó de interligação opera do seguinte modo. Quando a SCU recebe um pacote de controlo, converte-o de sinal óptico em sinal electrónico, processa a mensagem de *setup*, identifica o destino pretendido e consulta a tabela de encaminhamento para escolher o porto de saída. Se o porto pretendido estiver disponível no instante em que chega o *burst*, a SCU configura a matriz de comutação para encaminhar o *burst*. Se o porto não estiver disponível, dependendo da técnica de resolução da contenção implementada na rede, ou é deixado cair o *burst* que

chega, ou o *burst* que está a ocupar o porto é segmentado ou deixado cair. Por outro lado, se o *burst* chega ao OXC antes do correspondente pacote de controlo, este é deixado cair. Após a reserva do canal de dados, a SCU regenera o pacote de controlo, converte o sinal electrónico para o domínio óptico e envia-o para o próximo nó [90, 95, 101].

Agregação em *bursts*

Em redes OBS, a agregação em *bursts* [64, 90, 93, 95, 102-104] consiste, basicamente, no processo de junção de pacotes de dados em *bursts* no nó fronteira de origem. Neste nó fronteira, os pacotes que se destinam ao mesmo nó fronteira de destino e que pertencem à mesma classe em termos de qualidade de serviço são agregados e enviados em *bursts* de acordo com a política de agregação em *bursts* utilizada. No nó fronteira de destino, os *bursts* são sequencialmente desagregados e enviados electronicamente para o seu destino final.

No processo de agregação dos *bursts* existem dois parâmetros que determinam o modo como os pacotes são agregados, isto é, o tempo máximo de espera e o tamanho mínimo do *burst*. Com base nestes dois parâmetros, os algoritmos de agregação em *bursts* podem ser classificados nas três seguintes categorias:

- Algoritmos baseados no tempo [102, 104];
- Algoritmos baseados no tamanho do *burst* [90];
- Algoritmos híbridos [63, 95, 105].

Nos algoritmos baseados no tempo, o contador de tempo é inicializado sempre que o sistema arranca ou imediatamente após o envio do último *burst*. Quando o tempo pré-definido expira, o assemblador gera um *burst* e envia-o com todos os pacotes que estejam no *buffer* nesse instante [102]. Este algoritmo tem a vantagem de garantir um tempo mínimo de atraso para o processo de agregação quando se verifica pouco tráfego de entrada, ao passo que quando se apresenta muito tráfego de entrada, pode gerar *bursts* muito grandes aumentando o seu atraso desnecessariamente. Na literatura, exemplos destes tipos de algoritmos são propostos por Dolzer [106, 107] e Cao *et al*. [104].

Os algoritmos baseados no tamanho do *burst* caracterizam-se pelo facto de um *burst* ser gerado quando o número de pacotes no *buffer* atingir um determinado valor (o chamado valor *threshold*) [90]. Utilizando esta política de agregação, todos

os *bursts* têm o mesmo tamanho quando entram na rede de interligação. Uma vantagem deste esquema verifica-se quando existe muito tráfego de entrada, pois o valor definido para tamanho do *burst* é atingido rapidamente, minimizando o atraso dos dados. Por outro lado, quando se verifica pouco tráfego de entrada, o atraso será maior até se atingir o número de pacotes necessário no *buffer*.

As soluções híbridas procuram conjugar as vantagens dos dois esquemas anteriormente referidos. Por exemplo, em [95, 105, 108], um *burst* é gerado e enviado quando o tamanho do *burst* é atingido ou quando o tempo expira. Assim, verifica-se um atraso menor no envio dos *bursts*. Outras soluções híbridas propostas na literatura são as apresentadas em [92, 93, 103, 109, 110].

Protocolos de Reserva de Recursos para Redes OBS

Para transmitir um *burst* sobre uma rede OBS é necessário implementar um protocolo de reserva para alocar os recursos, configurando os comutadores ópticos para esse *burst* em cada nó da rede [64]. A sinalização é efectuada num canal independente dos canais de dados. Por outro lado, *burst offset* é o intervalo de tempo, no nó de origem, entre o processamento do último bit de dados da mensagem de *setup* e a transmissão do primeiro bit do *burst* de dados. Nesta tese, os protocolos de reserva de recursos são apresentados seguindo como critério o modo de reserva de recursos. Assim, os protocolos são classificados em duas classes: reserva unidireccional ou reserva bidireccional.

Nos esquemas de reserva unidireccional o *burst* é enviado pouco tempo depois do pacote de controlo e o nó origem não espera por uma mensagem de confirmação do nó destino. Assim, o tamanho do *offset* está entre o tempo de transmissão e o atraso de ida e volta de uma mensagem de sinalização, reduzindo o tempo de transferência de dados entre origem e destino. Diferentes protocolos podem escolher diferentes tempos de *offset* dentro desta gama de valores. *Tell And Go* (TAG) [111], *Just-In-Time* (JIT) [74, 75], *JumpStart* [75, 82-86], *Horizon* [1, 54], *Just-Enough-Time* (JET) [57, 73, 81] e JIT$^+$ [80] são exemplos de protocolos de reserva unidireccional de recursos.

No protocolo TAG o nó origem envia o pacote de controlo e, imediatamente a seguir, envia o *burst*. Em cada nó intermédio, o *burst* de dados tem que sofrer um atraso igual ao tempo de processamento da mensagem de *setup* (que é transportada no pacote de controlo). Se um canal não pode ser reservado ao longo do caminho ingresso-egresso, então o nó que precede o canal bloqueado deixa cair o *burst* e este

é perdido. Para libertar os recursos utilizados numa ligação, é enviado um sinal ou um pacote de controlo para o efeito [111, 112]. Neste protocolo, cada *burst* necessita de ser armazenado temporariamente em cada nó intermédio enquanto espera pelo processamento da mensagem de *setup* e pela configuração da matriz de comutação do *cross-connect* óptico (OXC). O TAG apenas é praticável se o tempo de processamento da mensagem de *setup* e o tempo de configuração do comutador óptico forem muito curtos [113].

Na categoria dos esquemas de reserva bidireccionais, o *offset* é o intervalo de tempo definido pela transmissão da mensagem de controlo e a recepção da confirmação do destino. A principal desvantagem desta categoria é o grande tempo de *offset*, que provoca um grande atraso no envio de dados. Exemplos de protocolos nesta categoria incluem o protocolo *Tell And Wait* (TAW) [111] e o esquema proposto em [114] (chamado *Wavelength Routed OBS network* - WR-OBS).

No protocolo TAW o nó origem envia um pacote de pedido de ligação (com a mensagem de *setup*) ao longo do caminho até ao nó destino informando que necessita de transmitir um *burst*. Se todos os nós intermédios puderem aceitar a ligação, o pedido é aceite e o nó fronteira de origem transmite. Caso contrário, o pedido é recusado e o nó de origem tenta novamente. Em cada nó intermédio, quando chega a mensagem de *setup*, a SCU reserva um canal de dados livre no caminho de saída. Cada canal de dados fica dedicado ao *burst* até receber uma mensagem explícita para libertação dos recursos. O nó fronteira de destino envia uma mensagem de confirmação, no sentido inverso da ligação, com o objectivo de notificar o nó origem acerca do sucesso no estabelecimento de um caminho óptico virtual para o *burst* pretendido. Esta mensagem de confirmação chega ao nó origem, caso tenha sido encontrado um canal de dados em cada ligação intermédia. Todos os nós intermédios que pertencem ao caminho virtual apenas vão enviar a mensagem de confirmação se a configuração da matriz de comutação estiver terminada, caso contrário, este envio será atrasado. Quando o nó origem recebe a mensagem de confirmação, envia o *burst* pelo caminho óptico virtual reservado. Logo que a transmissão do *burst* tenha terminado, o nó origem transmite uma mensagem de libertação que irá libertar os recursos reservados ao longo do caminho [111, 112].

O esquema de reserva bidireccional, chamado WR-OBS, proposto por Düser e Bayvel [114] combina OBS com comutação de circuitos de alta velocidade. O WR-OBS necessita obrigatoriamente de uma confirmação extremo a extremo e pretende fornecer uma arquitectura escalável e com garantias de qualidade de serviço. Depois do processo de agregação do *burst*, é efectuado um pedido de canal de dados

extremo a extremo para a transmissão do *burst* entre os nós fronteira origem e destino. Logo que um canal livre é encontrado, o *burst* é afectado a esse canal e transmitido para a rede de interligação. Após a transmissão do *burst*, o canal de dados é libertado e pode ser reutilizado para novas ligações.

Dadas as limitações evidenciadas pelos protocolos de reserva bidireccional e a limitação crítica a apresentada pelo TAG, este estudo concentra-se nos restantes protocolos de reserva unidireccional de recursos considerados mais relevantes na literatura. Esses protocolos são os seguintes: JIT, JumpStart, JIT$^+$, JET e Horizon. A seguir, descreve-se com mais detalhe cada um deles. No ponto que se segue ao próximo, é descrito o protocolo de reserva de recursos unidireccional proposto nesta tese, designado por *Enhanced Just-in-Time* (E-JIT).

Protocolos de Reserva Unidireccional de Recursos

Este ponto descreve os protocolos de reserva unidireccional de recursos existentes na literatura e que foram seleccionados para o presente estudo, tendo em conta as observações efectuadas no ponto anterior. Como se mostra na Figura II, estes protocolos podem ser classificados, atendendo ao modo como a reserva dos canais de dados de saída é efectuada para os *bursts*, em reserva imediata ou com reserva atrasada. O JIT e o JIT$^+$ são protocolos de reserva unidireccional com reserva imediata de canal, enquanto que o JET e o Horizon são exemplos de esquemas de reserva unidireccional com reserva atrasada de canal. O protocolo JumpStart tem a possibilidade de ser implementado usando reserva imediata ou com reserva atrasada [62].

Nos protocolos de reserva imediata, o canal de saída de *bursts* do OXC é reservado para os *bursts* de entrada imediatamente após a chegada da respectiva mensagem de *setup*. No caso dos protocolos com reserva atrasada, os recursos do OXC são reservados para o *burst* apenas antes da chegada do primeiro *bit* desse *burst*. Usando protocolos de reserva imediata, os elementos do OXC não são reservados para futuros *bursts* e assim o escalonamento não tem lugar nos nós OBS. Daí resulta que a matriz do OXC seja bastante simples [115]. Caso contrário, o escalonamento dos canais seria mais complicado.

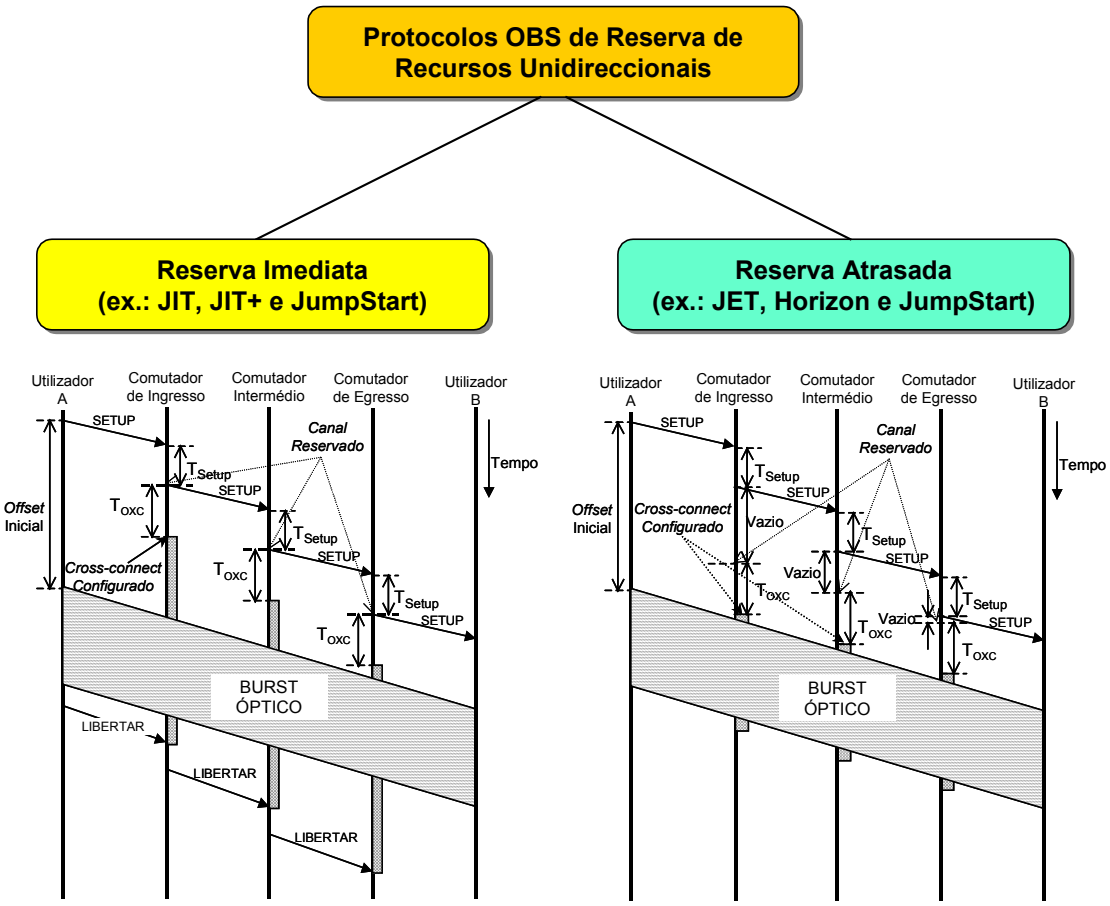


Fig. II. Classificação dos protocolos de reserva unidireccional de recursos para redes OBS.

O protocolo de reserva de recursos JIT foi proposto por Wei e McFarland [74]. Este protocolo considera que o canal de saída é reservado para um *burst* imediatamente após a chegada da correspondente mensagem de *setup*. Se um canal não pode ser reservado imediatamente, a mensagem de *setup* é rejeitada e o *burst* correspondente perde-se. O JIT usa libertação explícita dos recursos dos OXCs. Esta mensagem é enviada pelos nós fronteira (origem ou destino) para libertar os recursos de todos os OXC ao longo da ligação para envio do *burst*. Sempre que um nó detecte uma mensagem de falha, envia uma mensagem de libertação para todos os nós ao longo do caminho até ao nó origem. A Figura II ilustra o funcionamento do protocolo JIT. Como se pode observar nesta figura, T_{Setup} representa o tempo necessário ao processamento da mensagem de *setup* no nó OBS, e T_{OXC} representa o atraso decorrido entre o instante em que o OXC recebe o comando da SCU para definir uma ligação de um porto de entrada para um porto de saída, e o instante em que um caminho adequado fica definido na rede OBS e esta pode ser usada para enviar um *burst* [80].

O JumpStart [75, 82-85] é um projecto conjunto apoiado pela *Advanced Research and Development Agency* (ARDA) desenvolvido pela Universidade do Estado da Carolina do Norte e pelo *MCNC Research and Development Institute*. O objectivo deste projecto é a definição de um protocolo de sinalização e a respectiva arquitectura associada para redes de comutação de agregados de pacotes (redes OBS) usando multiplexagem por divisão no comprimento de onda (WDM). No protocolo JumpStart [75, 115], um nó fronteira de origem, primeiro envia uma mensagem de *setup* para o nó de ingresso a ele ligado com informação relativa à transmissão de um *burst*, incluindo os endereços origem e destino. Se o nó de ingresso pode comutar o *burst*, envia uma mensagem de *setup ACK* (*acknowledgement*) para o nó fronteira origem e envia também a mensagem de *setup* para o nó seguinte. Caso contrário, o nó de ingresso recusa a mensagem de *setup*, envia uma mensagem de rejeição para o nó fronteira de origem e o *burst* correspondente perde-se. Neste caso, o nó fronteira entra no estado inactivo esperando por um novo *burst*. Quando um novo *burst* chega, o nó fronteira repete o processo.

Nos esquemas de reserva atrasada, quando um *burst* é aceite por um nó OBS, o canal de saída é reservado por um período de tempo igual ao tempo de transmissão do *burst* mais o T_{OXC}, para prever o tempo necessário à configuração do OXC. Como se pode ver na Figura II, que apresenta a operação do protocolo Horizon, é criado um vazio no canal de saída entre o tempo $t + T_{Setup}$ (quando a operação de reserva para o *burst* que vai chegar fica terminada), e o tempo $t' = t + T_{offset} - T_{OXC}$, quando o comprimento de onda de saída é realmente reservado para o *burst*. O protocolo Horizon é um exemplo dos protocolos com reserva atrasada sem preenchimento de vazios, e é menos complexo que os protocolos com reserva atrasada e preenchimento de vazios, como, por exemplo, o JET.

O protocolo Horizon foi proposto por Turner no âmbito do projecto *Terabit Burst Switching* [1, 54]. O Horizon é considerado um protocolo de reserva de recursos no sentido em que incorpora uma reserva com atraso, tal como é referido em [75, 80, 82]. Este protocolo introduz o conceito de horizonte temporal (*time horizon*) para efeitos de reserva de um determinado canal e é chamado *Horizon* porque cada canal de dados tem o seu horizonte temporal para o qual está reservado. O horizonte temporal é definido como "o tempo mais cedo para o qual não há previsão de uso do canal (comprimento de onda)" [1]. Este conceito é usado em outros protocolos de reserva unidireccional de recursos como são o caso do JET e do JIT$^+$. No Horizon, um canal de saída é reservado para um *burst* se a chegada desse *burst* é posterior ao

horizonte temporal desse canal. Se, à chegada de uma mensagem de sinalização, se verifica que o tempo de chegada do *burst* é anterior ao menor horizonte temporal para qualquer canal disponível, então essa mensagem é rejeitada e o *burst* correspondente perde-se [1].

O protocolo de reserva de recursos JET foi proposto em [57]. Neste protocolo, um canal é reservado para um *burst* se o tempo de chegada desse *burst* (*i*) é posterior ao horizonte temporal desse canal, ou (*ii*) coincide com um vazio desse canal e o fim do *burst* (mais o tempo de configuração do OXC - T_{OXC}) ocorre antes do fim do vazio. Se à chegada da mensagem de sinalização se verifica que nenhuma destas condições é satisfeita para um qualquer canal disponível, então essa mensagem é rejeitada e o *burst* correspondente perde-se. O JET é o protocolo de reserva de recursos com preenchimento de vazios mais conhecido, e usa informação (existente na mensagem de *setup*) para prever o início e o fim do *burst*. Os autores do JET fizeram estudos analíticos e de simulação que confirmaram o efeito positivo da reserva atrasada na probabilidade de perda de *bursts* numa rede OBS.

O mais recente protocolo de reserva de recursos proposto é o JIT$^+$ [80]. No JIT$^+$, um canal de saída é reservado para um *burst* se (*i*) o tempo de chegada do *burst* é posterior ao horizonte temporal desse canal, e (*ii*) se o canal tem no máximo uma outra reserva. O JIT$^+$ foi definido como um melhoramento do JIT e combina a simplicidade do JIT com a utilização do horizonte temporal usada pelos protocolos de reserva atrasada, como o Horizon ou o JET. O JIT$^+$ é uma versão modificada do protocolo JIT que adiciona um escalonamento de *bursts* limitado (com um máximo de dois *bursts* por canal). Este protocolo não realiza nenhum tipo de preenchimento de vazios nos tempos de reserva. Comparando o JIT$^+$ com o JET e com o Horizon, os últimos permitem um número ilimitado de reservas por comprimento de onda, enquanto que o JIT$^+$ limita o número destas reservas por canal a, no máximo, uma para além da reserva actual.

Novo Protocolo de Reserva de Recursos: *Enhanced JIT* (E-JIT)

Nesta tese é proposto um novo protocolo de reserva unidireccional de recursos, designado por *Enhanced Just-in-Time* (E-JIT). Esta proposta baseia-se na avaliação de desempenho relativa aos protocolos JIT, JumpStart, JIT$^+$, JET e Horizon e a sua complexidade relativa (o protocolo JIT é o mais fácil de implementar). O protocolo E-JIT pretende melhorar e optimizar o tradicional JIT, mantendo todas as

suas vantagens em termos de implementação. O E-JIT visa melhorar o escalonamento e a utilização do canal de dados, reduzindo o período de tempo no qual o canal está no estado "reservado" e os seus períodos de tempo livres (até zero, caso seja possível), optimizando a sua utilização e potencialmente reduzindo a probabilidade de perda de *bursts*.

Utilizando o E-JIT, a reserva de um canal de dados é efectuada para um *burst* imediatamente após a chegada da sua mensagem de *setup* (*i*) se este canal de dados estiver livre ou (*ii*) se, caso o canal de dados esteja ocupado, o tempo de fim de *burst* (*end time*) do último *burst* comutado for menor do que o tempo actual para processar a mensagem de *setup* ($\leq T_{Setup}$). Se o canal não puder ser reservado para o *burst*, então a mensagem de *setup* é rejeitada e o correspondente *burst* é deixado cair.

Tanto o JIT como o E-JIT são protocolos de reserva de recursos com reserva imediata. O E-JIT assume uma sinalização com um canal associado, suportando tanto *bursts* longos como curtos. A libertação dos recursos alocados para a transmissão de determinado *burst* é efectuada de forma implícita ou estimada em função do instante de chegada e da duração do respectivo *burst*. Neste aspecto, o E-JIT é semelhante ao JET.

Resolução da Contenção

Usando protocolos de reserva unidireccional, o nó origem envia *bursts* de saída sem receber uma mensagem de confirmação ou uma coordenação global [63]. Assim, nos nós intermédios o problema da possível contenção entre *bursts* tem que ser resolvido.

A contenção ocorre quando mais que um *burst* disputa simultaneamente o mesmo canal de saída [93]. Em redes OBS, este problema é agravado pelo tamanho variável dos *bursts* e pela existência de *bursts* de longa duração [116]. Na comutação electrónica de pacotes, a contenção é resolvida com o recurso a *buffers*. No entanto, no domínio óptico não existem *buffers* equivalentes para resolver este problema.

Quando a contenção ocorre, o *burst* que a provoca pode ser simplesmente deixado cair, ou deixado cair e posteriormente retransmitido [117]. Se é usada uma política de deixar cair os *bursts*, o *burst* que provoca a contenção é simplesmente rejeitado e perde-se. Quando isto se verifica, o nó que o rejeita envia uma mensagem de falha (*negative acknowledgement*) para o nó origem, que por sua vez

notifica o *router* IP origem. O nó origem não retransmite o *burst* porque, utilizando esta política, a retransmissão é inteiramente gerida pelo *router* IP origem ou pela aplicação que gerou o pedido.

A política de retransmissão é semelhante à política de deixar cair os *bursts* excepto na retransmissão que é feita pelo nível WDM. Após a recepção de uma mensagem de falha, o comutador (*switch*) origem retransmite o *burst* rejeitado sem notificar o *router* IP origem. Contudo, para melhorar a probabilidade de perda de *bursts* e para melhorar o desempenho das redes OBS existem quatro técnicas principais para a resolução da contenção. Essas técnicas são as seguintes: *buffers* ópticos usando linhas de atraso baseadas em fibras (*fiber delay lines* - FDLs) [44, 46, 47, 57, 118], conversão do comprimento de onda [119-122], encaminhamento por deflexão (*deflection routing*) [46, 57, 123-125], e segmentação dos *bursts* [93, 116, 126-128].

As linhas de atraso baseadas em fibras (FDLs) [44, 46, 47, 57, 118] (deflexão aplicada no domínio do tempo [63, 129, 130]) são a solução conhecida para implementar *bufferização* óptica. Uma FDL pode atrasar um sinal óptico uma determinada unidade de tempo, de acordo com o comprimento da fibra óptica. Um conjunto de FDLs pode fornecer diversas unidades de tempo de atraso em função do número de fibras, sendo que cada fibra fornece mais uma unidade de tempo de atraso em relação à anterior. Contrastando com os *buffers* electrónicos, as FDLs apenas fornecem um atraso fixo e usam o princípio do primeiro a chegar é o primeiro a sair, no sentido em que os *bursts* deixam as FDLs pela mesma ordem pela qual entraram. Quando as FDLs são utilizadas numa rede OBS representam um custo adicional em termos de hardware.

A conversão do comprimento de onda [119-122] (deflexão aplicada no domínio do comprimento de onda [63, 129, 130]) consiste na conversão do comprimento de onda de um canal de entrada noutro comprimento de onda do canal de saída. Quando um canal de saída está ocupado com um *burst* e surge um novo *burst* para o mesmo destino, recorrendo a esta técnica de resolução da contenção, o comprimento de onda do canal do segundo *burst* é convertido e o *burst* é enviado através de outro comprimento de onda disponível. Assim, resolve-se o problema da contenção no canal de saída de *bursts*.

O encaminhamento por deflexão foi proposto para a resolução da contenção no contexto das redes OBS por Wang *et al*. [123] em 2000. Usando esta técnica [46, 57, 123-125] (deflexão usada no domínio do espaço [63, 129, 130]), o *burst* que provoca a contenção é encaminhado para um porto alternativo e, assim, segue um

caminho alternativo até ao seu destino. Desta forma, seguindo um caminho alternativo, o *burst* sofre um atraso maior do que se tivesse seguido pelo caminho mais curto. No entanto, quando se escolhe um caminho alternativo para deflexão, procura-se escolher o caminho (alternativo) mais curto para o destino. Se não houver portos alternativos disponíveis, o *burst* é descartado [117]. Um inconveniente da deflexão é a possibilidade do *burst* entrar em ciclo infinito (*looping*) caso seja sempre encaminhado pelo mesmo caminho alternativo sem encontrar o seu nó destino.

A segmentação dos *bursts* [93, 116, 126-128] foi introduzida para reduzir a probabilidade de perda de *bursts* em redes OBS [116, 126]. Esta técnica segue o princípio que é melhor dividir o *burst* em diversos segmentos e apenas deixar cair os segmentos sobrepostos que perder o *burst* completo durante a contenção. Quando a contenção ocorre, os *bursts* são divididos em segmentos e apenas alguns segmentos de um dos *bursts* envolvidos podem ser perdidos. Inúmeras técnicas baseadas na segmentação de *bursts* são propostas na literatura para resolver o problema da contenção, podendo ser encontradas em [116, 126, 127, 131, 132]. Como exemplo, Vokkarane *et al*. [116, 126] consideram duas abordagens principais para deixar cair segmentos do burst:

1. *Tail-dropping* – deixar cair a cauda do *burst* original, ou
2. *Head-dropping* - deixar cair a cabeça do burst que causa a contenção.

Modelo de Tráfego dos Agregados de Pacotes

Como foi mencionado anteriormente aquando da descrição dos nós de interligação OBS, assume-se que cada nó necessita de [80, 133]: *i*) uma quantidade de tempo, T_{OXC}, para configurar a matriz de comutação do OXC no sentido de estabelecer uma ligação entre um porto de entrada e um porto de saída e *ii*) de um período de tempo, $T_{Setup}(X)$ para processar a mensagem de *setup* para o protocolo de reserva de recursos X, onde X pode representar o JIT, JumpStart, JIT$^+$, JET, horizon, ou o E-JIT. Considera-se também o valor do *offset* de cada *burst* para cada protocolo de reserva de recursos X, $T_{offset}(X)$, que depende, entre outros factores, do protocolo de reserva de recursos utilizado, do número de nós que o *burst* já atravessou, e se o valor do *offset* é usado para diferenciação de serviços. Assume-se que o número de saltos (*hops*) no percurso de cada *burst* numa dada rede é

uniformemente distribuído desde o nó de interligação de ingresso até ao nó de interligação de egresso, e o tempo de *offset* mínimo do *burst* ($T_{offset}(X)$) é calculado por:

$$T_{offset}^{(\min.)}(X) = k.T_{Setup}(X) + T_{OXC} \qquad \text{(EA.1)}$$

onde *k* representa o número de nós OBS desde o nó origem até ao nó destino da ligação.

Este estudo segue as condições de simulação apresentadas em [80, 133] em termos dos cenários de tráfego. Na obtenção dos resultados de simulação, para estimar a probabilidade de perda de *bursts*, foi estimado um intervalo de confiança de 95% usando o método de *batch means* [134, 135]. O número de *batches* é 30 (o valor mínimo para obter o intervalo de confiança) e cada *batch* é executado até garantir que pelo menos 2560000 *bursts* são transmitidos, assumindo que cada nó fronteira transmite, pelo menos, 200 *bursts* e cada *batch* contém 10 observações. O valor de, pelo menos, 2560000 *bursts* transmitidos é obtido para uma rede com *N*=20 nós, i.e., 200 *bursts* transmitidos x 64 nós fronteira x 20 nós de interligação x 10 observações = 2560000 *bursts* transmitidos. Verificou-se que os intervalos de confiança são muito pequenos. Deste modo, neste estudo não são apresentados os intervalos de confiança nas figuras com o objectivo de aumentar a sua legibilidade.

No que concerne ao número de canais de dados (*F*) disponíveis em cada ligação, assume-se que o número de canais ligando dois nós é *F*+1. *F* é o número de canais de dados e o outro representa o canal de sinalização. Neste estudo, *F* pode ter o valor de 16, 32, 64 ou 128 canais de dados por ligação ($F=2^n$, com $4 \leq n \leq 7$). Todos os canais são bi-direccionais.

Em termos do processo de chegada das mensagens de *setup* (e, como consequência, dos *bursts* de dados), é assumida uma distribuição de Poisson com taxa λ, tal como é utilizado em [1, 3-6, 80, 115, 133, 136]. O tamanho do *burst*, tanto curto como longo, segue uma distribuição exponencial com um tamanho médio de *burst* de $1/\mu$, como em [80, 93, 94, 115, 116, 137, 138]. Como em [80, 93, 94, 115, 116, 137, 138], assume-se que o comprimento do *burst*, quer seja curto ou longo, é limitado [43] e segue uma distribuição exponencial com um tamanho médio de *burst* de $1/\mu$. Deste modo, no OBSim, tendo em conta a distribuição do tamanho médio do *burst* ($1/\mu$) e a taxa de chegada da mensagem de *setup* λ, o rácio da geração de *bursts* é representado por λ/μ. Também se assume que os *bursts* são enviados uniformemente para todos os nós de interligação da rede, com a excepção que um nó não pode enviar mensagens para ele próprio e um nó de interligação pode

gerar, pelo menos, uma mensagem por *time-slot* ou período de tempo. Os nós fronteira são responsáveis pelo processo de geração dos *bursts*, isto é, nem os nós de interligação de ingresso nem outros quaisquer nós de interligação processam o *burst*.

Outro aspecto importante relacionado com este estudo diz respeito ao número de conversores de comprimento de onda em cada nó. Assume-se que cada nó OBS suporta conversão óptica de todos os comprimentos de onda. Em termos do número de nós fronteira ligados a cada nó de interligação, assume-se que, para efeitos de simulação, eles são uniformemente distribuídos e cada nó de interligação tem ligados 64 nós fronteira. Entre os nós de interligação assume-se que a distância geográfica é longa de tal forma que o atraso típico em cada ligação é da ordem dos 10ms [139].

Durante a simulação, para seleccionar um canal de dados livre para um *burst* de entrada (com igual probabilidade), para o JIT, JumpStart e JIT$^+$ é usada a política de atribuição do comprimento de onda aleatório [140], e para o JET e o Horizon é usado o algoritmo do *último canal disponível não utilizado* (*latest available unused channel* - LAUC) [95].

Desenho e Construção de uma Ferramenta de Simulação para Redes OBS

A tecnologia OBS levantou um número significativo de questões, como a análise de diferentes protocolos de reserva de recursos e a relação destes protocolos com diferentes perfis de utilização e topologias de rede, considerando diferentes cenários em termos de tráfego por nó. Estas questões podem ser respondidas com recurso a um simulador que reproduza o comportamento de uma rede OBS, dada a inexistência destas redes no mundo real. No entanto, existem algumas redes OBS em laboratórios de investigação e redes de ensaio que são reportadas em [75, 141-143].

Trabalhos anteriores em simuladores para redes ópticas baseiam-se em tráfego de pacotes (ex.: redes IP) que são significativamente diferentes do tráfego de agregados de pacotes (*bursty*) das redes OBS, uma vez que os *bursts* são transmitidos de uma forma transparente na rede, no sentido em que esta rede não reconhece nem o fim nem o conteúdo dos *bursts*. Nesse sentido, torna-se necessário construir novas ferramentas de simulação com o objectivo de incluírem estas características específicas do tráfego de agregados de pacotes. Assim, foi proposta

uma abordagem orientada a objectos para a construção de uma ferramenta de simulação para redes OBS, designada por OBSim, utilizando a linguagem Java. Este simulador suporta os protocolos de reserva de recursos estudados nesta tese (JIT, JumpStart, JIT$^+$, JET e Horizon), bem como a proposta de um novo protocolo – o E-JIT -, e pode simular diferentes cenários de tráfego e diferentes topologias de rede definidas pelo utilizador.

O OBSim está desenhado para construir um modelo de uma rede OBS baseada em objectos, com os seguintes objectivos:

- Comparar o desempenho dos diferentes protocolos de reserva de recursos com base na probabilidade de perda de *bursts*;
- Estudar a influência dos diferentes parâmetros de rede no desempenho das redes OBS;
- Avaliar o desempenho das redes OBS para diferentes topologias definidas pelo utilizador;
- Comparar o desempenho do OBS com outras tecnologias;
- Testar novos protocolos de reserva de recursos.

No que concerne à simulação de redes, as ferramentas de investigação enquadram-se em três categorias diferentes: modelos analíticos, medidas *in situ* e simuladores [144]. O OBSim é um simulador despoletado por acontecimentos, estocástico e simbólico. No OBSim, os acontecimentos que correm no simulador são mensagens. Estas podem ser enviadas pelos nós fronteira ou geradas pelos nós de interligação, de acordo com o protocolo de reserva de recursos utilizado. Os simuladores estocásticos, em oposição aos determinísticos, confiam em entidades aleatórias (geralmente variáveis aleatórias de valor numérico) para simular a aleatoriedade dos eventos da vida real. No OBSim, usa-se a classe Java *Random*, que gera valores pseudo-aleatórios (de diversos tipos) e existem dois testes que uma variável pseudo-aleatória tem que passar [135]. O primeiro é a homogeneidade da distribuição e o segundo é o da independência dos valores gerados. A classe Java *Random* satisfaz estas condições [135, 145]. Os simuladores simbólicos usam alguns tipos de símbolos para copiarem o comportamento dos elementos reais. Assim, no OBSim, os símbolos são as classes Java, que são instanciadas quando necessário pelo software, de acordo com os dados de entrada fornecidos inicialmente pelo utilizador.

A geração de tráfego é um aspecto importante neste modelo. Como o OBSim é um simulador despoletado por acontecimentos, inicialmente há que simular a

necessidade de transmissão de *bursts* entre nós. Em cada simulação, assume-se que os *bursts* são enviados uniformemente para todos os nós da rede. Como cada *burst* tem que ser precedido por uma mensagem de *setup* e os nós fronteira ligados aos nós de interligação enviam *bursts* aleatoriamente no tempo, considera-se que o tempo entre estas mensagens segue uma distribuição exponencial, simples ou com um tempo de *offset* [146, 147]. O tráfego é então simulado quando o OBSim começa a processar a fila de espera de acontecimentos, que, quando o programa arranca, está carregada com os pedidos (mensagens) dos nós fronteira. Estes pedidos, quando são processados, geram normalmente mais mensagens que são adicionadas à fila de espera. Cada vez que uma mensagem é adicionada à fila de espera, o temporizador do simulador gera um intervalo de tempo de acordo com a distribuição definida pelo utilizador, e este tempo é adicionado ao relógio do simulador, definindo o instante em que este acontecimento será escalonado para ocorrer.

Em termos de execução, o OBSim segue um esquema funcional linear, desempenhando cinco tarefas base que são as seguintes: (1) validação dos parâmetros iniciais; (2) leitura da topologia de rede; (3) construção do modelo de rede; (4) simulação do tráfego; e (5) apresentação dos resultados.

A validação dos valores iniciais permite definir, em conjunto com a topologia de rede o modelo abstracto da rede que se pretende simular. Cada nó da rede, de acordo com os parâmetros de carga e do tipo de carga, é depois instado a gerar mensagens que são colocadas numa fila de espera. A topologia de rede, depois de lida do respectivo ficheiro de configuração, é usada para criar o modelo abstracto da rede. Depois de criados os modelos da rede usando as classes disponíveis no simulador, e de inicializadas as estruturas de dados de cada uma, a classe `Routing` vai executar o algoritmo de Dijkstra para cada um dos nós. Os caminhos mais curtos assim conseguidos são armazenados na classe `Routing`, num mapa. Em seguida, a classe `Simulator` inicializa cada nó de interligação, e para cada um destes nós, faz arrancar cada um dos seus nós fronteira (se estes forem transmissores). O processo de arranque de cada nó fronteira, codificado na classe `Edge`, consiste essencialmente em adicionar um pedido de envio de um *burst* ao calendarizador de eventos (classe `EventCalendar`). Depois de todos os nós fronteira que tiverem permissão para emitir colocarem o seu pedido de envio de um *burst*, o OBSim vai realizar todos os eventos que estiverem pendentes no calendarizador. Esta realização de eventos, que pode ser executada ou numa instância da classe `Node` ou numa instância da classe `Edge`, gera mais eventos que vão sendo adicionados à fila de espera. O simulador termina quando já não há mais eventos para realizar. Nesta

altura é chamado um método que calcula os rácios dos valores entretanto armazenados nas várias classes. Estes valores são, de forma resumida, o número de *bursts* enviados e perdidos em cada nó de interligação e em cada nó fronteira, e o número de *bursts* enviados e perdidos por salto por nó, em cada nó e em cada utilizador (classes `Edge` e `Node`). Com estes valores, é calculada a taxa de perda de *bursts* em cada salto (*hop*) da rede.

A interface com o utilizador do simulador permite a definição de diversos atributos. A rede a simular é definida num ficheiro de texto no qual se definem, nomeadamente, os nós e as ligações entre eles, o número de nós fronteira em cada nó e o tempo de atraso na propagação do sinal entre nós de interligação. Os parâmetros de simulação definidos através da interface com o utilizador são os seguintes:

1. *Protocolo de Reserva de Recursos (Resource reservation protocol*) – Neste campo selecciona-se o protocolo de reserva de recursos a utilizar. Os valores possíveis são JIT, JumpStart, JIT$^+$, JET, Horizon e E-JIT.
2. Função de Geração da Distribuição (*Generation distribution function*) – Este campo define o modelo estatístico para gerar o intervalo de tempo entre acontecimentos no simulador. As funções permitidas são a Exponencial e a Exponencial de Dois Estados.
3. Rácio da Geração de *Bursts* (*Burst generation ratio*) – Este campo admite um valor real entre 0 e 1 e representa a taxa de geração de *bursts* por nó (λ/μ[4]).
4. Número de Canais de Dados Disponíveis por Ligação (*Available data channels per link*) – Este campo admite valores numéricos entre 1 e 2147483647[5] e representa o número de canais de dados disponíveis por ligação.
5. Tempo de Processamento da Mensagem de *Setup* (*Setup message process time*) – Este campo admite um valor decimal maior que 0 e representa o tempo que o OXC necessita para processar a mensagem de *setup*, definido como T_{Setup}.
6. Tempo para Configuração do Comutador (*Switch configuration time*) - Este campo admite um valor decimal maior que 0 e representa o tempo

[4] λ/μ - em que é assumida uma distribuição de Poisson com taxa λ, e $1/\mu$ representa o tempo médio de duração do *burst*.
[5] Os valores apresentados correspondem ao intervalo positivo não nulo de valores possível no tipo `int` da linguagem Java.

que o comutador necessita para configurar a matriz de comutação, depois de receber a mensagem de *setup* e a interpretar, definido como T_{OXC}.

7. Tempo de Atraso entre o Nó Fronteira e o seu Nó de Interligação (*Edge to core node delay*) – Este é um campo numérico maior que 0 e guarda o tempo que a mensagem de *setup* (e os *bursts*) demora entre o nó fronteira e o seu nó de ingresso/egresso.
8. Ficheiro com a Topologia da Rede (*Network topology file*) - Este campo admite um valor alfanumérico que represente o caminho (completo ou relativo) para o ficheiro com a definição da topologia da rede. Tem associado um botão (`file...`) que permite visualizar uma listagem dos ficheiros disponíveis.

A validação é um ponto chave para se poder confiar na utilização dos resultados fornecidos por qualquer simulador. Perros [135] define a validação de um modelo através da verificação das seguintes cinco etapas: 1) verificar o gerador de números pseudo-aleatórios; 2) verificar o gerador de variáveis estocásticas; 3) verificar a lógica do programa de simulação; 4) validade das relações; e 5) validade dos resultados.

No OBSim, a exactidão do gerador de números pseudo-aleatórios é garantida pelos padrões da definição da linguagem Java, e confirmada pelo teste do Qui-Quadrado e pelo teste da independência dos valores gerados, satisfeitos pela classe Java *Random* [135, 148]. O gerador de variáveis estocásticas foi validado separadamente em [147, 149, 150]. A lógica do programa de simulação e a validade das relações são inerentes aos protocolos de reserva de recursos e ao ambiente de programação em Java. A validade dos resultados foi efectuada através da comparação com os resultados publicados em [80]. Com este objectivo, foi executado um exemplo de simulação considerando um único nó OBS de interligação isoladamente, para os protocolos de reserva de recursos JIT, JET e Horizon. São utilizados os mesmos parâmetros de carga/tráfego apresentados em [80]: T_{Setup}(JIT)=12.5μs, T_{Setup}(JET)=50μs, T_{Setup}(Horizon)=25μs, T_{OXC}=10ms, o tempo médio de duração do *burst* $1/\mu$ foi ajustado a 50ms (igual a $5T_{OXC}$), e a taxa de chegada λ das mensagens de *setup* (e dos *bursts*) é tal que λ/μ=32, assumindo que estão 64 nós fronteira ligados ao nó de interligação.

A Figura III mostra a probabilidade de perda de *bursts* em função do número de canais de dados por ligação para um único nó OBS de interligação isoladamente, dada pelo OBSim e comparada com os resultados apresentados em [80]. Como se pode observar na figura, os resultados obtidos pelo OBSim são muito semelhantes aos publicados em [80]. A pequena variação perceptível é espectável, dada a natureza estocástica dos acontecimentos que estão a ser modelados.

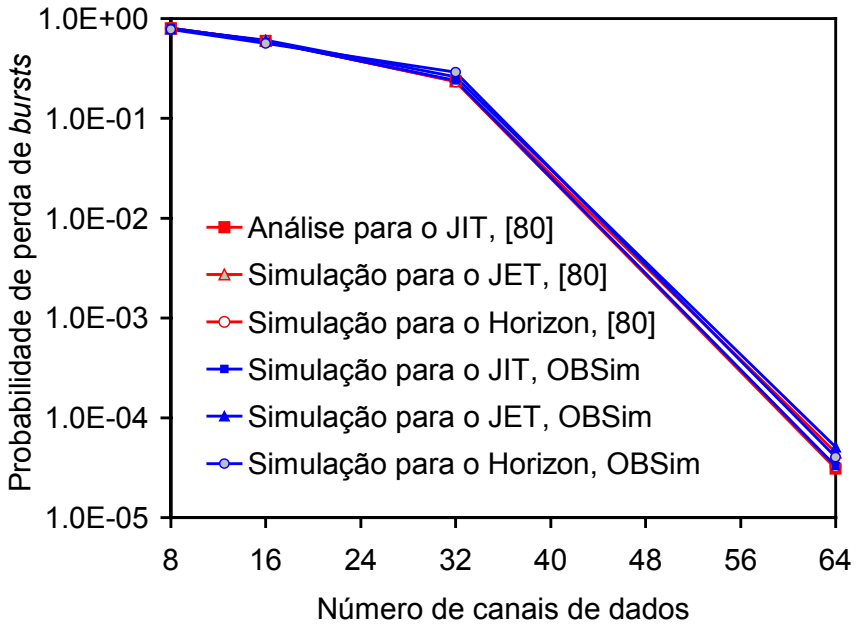


Fig. III. Probabilidade de perda de *bursts*, em função do número de canais de dados por ligação (*F*) [18], num único nó para os protocolos de reserva de recursos JIT, JET e Horizon dada pelo OBSim comparada com os resultados publicados em [80].

Topologias de Redes de Interligação

A topologia de rede refere-se à forma como os diferentes nós estão interligados entre si, e o modo como comunicam é determinado por esta mesma topologia. Em todas as topologias de redes OBS utilizadas neste estudo considera-se que as ligações entre nós são bidireccionais, ou seja, o tráfego flui em ambos os sentidos.

Diversas topologias foram consideradas para avaliar o desempenho dos protocolos de reserva unidireccional de recursos estudados. Foram consideradas tanto topologias regulares como irregulares. Como topologias regulares foram

consideradas as seguintes: anéis, anéis com cordas (de grau nodal 3, 4, 5 e 6) e teia toroidal (com 16 nós e 32 ligações, e com 25 nós e 50 ligações). No que concerne a topologias irregulares, consideram-se as seguintes: NSFNET (com 14 [64, 151] e 16 nós [152]), ARPANET [151, 153], rede óptica europeia (*European Optical Network* - EON) [154] e uma proposta de rede para a Fundação para a Computação Científica Nacional (FCCN) (designada por FCCN-NET).

Os anéis com cordas são uma família bem conhecida de topologias de grau três propostas por Arden e Lee no princípio dos anos oitenta para interligação de sistemas multi-computador [155]. Um anel com cordas é uma rede em anel bidireccional, na qual cada nó tem uma ligação bidireccional adicional, chamada corda. O número de nós de um anel com cordas deve ser par, e os nós são indexados sequencialmente de *0* até N-1, à volta do anel com N nós. Também se define que cada nó de índice ímpar i (i=1, 3, …, N-1), está ligado a um nó de índice $(i+w) mod\ N$, onde w é o comprimento da corda, que é assumido como sendo um ímpar positivo. Para um dado número de nós, existe um comprimento de corda óptimo que conduz ao menor diâmetro de rede. O diâmetro de rede é o maior de todos os caminhos mais curtos entre todos os pares de nós, sendo o comprimento do caminho definido pelo número de saltos (*hops*) desse caminho. Em cada nó de um anel com cordas, tem-se uma ligação ao nó anterior, outra ligação ao nó seguinte, e uma corda. Aqui, é assumido que as ligações aos nós anterior e seguinte são substituídas por cordas. Assim, cada nó tem três cordas, em vez de uma. Sejam w_1, w_2, e w_3 os seus respectivos comprimentos e N o número de nós. Representamos uma topologia geral de grau três por D3T(w_1, w_2, w_3). Assumimos que cada nó de índice ímpar i (i=1, 3, …, N-1) está ligado aos nós $(i+w_1) mod\ N$, $(i+w_2) mod\ N$, e $(i+w_3) mod\ N$, onde os comprimentos das cordas w_1, w_2, and w_3 são ímpares positivos, com $w_1 \leq N$-1, $w_2 \leq N$-1, e $w_3 \leq N$-1, e $w_i \neq w_j$, $\forall_{i \neq j} \wedge 1 \leq i,j \leq 3$. Nesta notação, um anel com cordas com comprimento de corda w_3 é representado por D3T(1,N-1,w_3).

Em seguida, apresenta-se a topologia geral para qualquer grau nodal. Assume-se que em vez de uma topologia de grau nodal três, se tem uma topologia de grau nodal n, onde n é um inteiro positivo, e em vez de três cordas, ter-se-ão n cordas. É assumido que cada nó de índice ímpar i (i=1,3,…,N-1) está ligado aos nós $(i+w_1) mod\ N$, $(i+w_2) mod\ N$, …, $(i+w_n) mod\ N$, onde os comprimentos das cordas w_1, w_2, … w_n são ímpares positivos, com $w_1 \leq N$-1, $w_2 \leq N$-1, …, $w_n \leq N$-1, e $w_i \neq w_j$, $\forall_{i \neq j} \wedge 1 \leq i,j \leq n$. Agora introduz-se uma nova notação: uma topologia de grau n é representada por

DnT(w_1,w_2,…,w_n). Nesta notação, uma família de anéis com cordas com comprimento de corda igual a w_3 é representado por D3T(1,N-1,w_3) e um anel bidireccional é representado por D2T(1,N-1).

Nesta tese é também apresentada uma proposta para a topologia de *backbone* de rede para a Fundação para a Computação Científica Nacional (FCCN) [156] sobre a estrutura da REFER Telecom S.A. (REFER Telecom) [157], empresa que possui capacidade de implementação para uma infra-estrutura em fibra óptica (Figura IV). O objectivo desta proposta é estudar o desempenho desta rede, designada por FCCN-NET, usando a tecnologia OBS, considerando apenas um conjunto de nós significativos do ponto de vista de tráfico da FCCN, isto é, ligações a universidades e a pólos universitários.

Fig. IV. Topologia proposta para a rede da FCCN (com 14 nós e 14 ligações).

A rede é formada por 14 nós e 14 ligações, dos quais 13 são propriedade da REFER Telecom, e o décimo quarto nó é propriedade da FCCN (GigaPix de Lisboa), existindo uma ligação em fibra óptica entre o Gigapix de Lisboa e o nó da REFER Telecom na Gare do Oriente. Nesta rede são considerados cinco nós principais (Lisboa, Porto, Coimbra, Aveiro e o GigaPix de Lisboa), oito nós secundários (Braga, Vila Real, Guarda, Covilhã, Castelo Branco, Setúbal, Évora e Faro) e um nó de interligação (Entroncamento). A diferença entre os nós principais e secundários reside na quantidade de tráfego local (de e para os nós fronteira). Assume-se que os nós secundários têm metade do tráfego local em termos de probabilidade da distribuição do tráfego. O nó do GigaPix de Lisboa é assumido como a porta para o

tráfego internacional, embora a REFER Telecom possa ter outros pontos de ligação internacionais.

Avaliação do Desempenho de Redes OBS com Protocolos de Reserva Unidireccional de Recursos

Este ponto é dedicado à avaliação do desempenho dos protocolos de reserva unidireccional de recursos JIT, JumpStart, JIT$^+$, JET e Horizon e o protocolo proposto, E-JIT, para redes OBS com topologias em malha. Este estudo de avaliação de desempenho pode ser expresso em termos da probabilidade de perda de *bursts* obtida a partir do simulador descrito anteriormente.

Usando protocolos de reserva unidireccional de recursos, a perda de *bursts* pode ocorrer em redes OBS porque os pacotes de controlo não conseguem reservar os recursos em algum dos nós intermédios. Outro factor importante é a possibilidade de perda de *bursts* caso o próprio canal de controlo esteja congestionado ou se verifique outra falha neste canal. Deste modo, a probabilidade de perda de *bursts* é uma métrica importante no desempenho das redes OBS [60]. Quando se perde um *burst*, a sua retransmissão é deixada à responsabilidade dos protocolos das camadas superiores.

Na revisão da literatura, existem diversos estudos de avaliação do desempenho em redes OBS, nomeadamente, em [4, 80, 113, 133, 137, 158-162]. Assim, esta tese considera apenas a existência de redes sem armazenamento temporário de dados (ex.: nós intermédios sem linhas de atraso baseadas em fibras) e o estudo concentra-se na perda de *bursts* em redes OBS em malha. Como resultado, sempre que um *burst* não pode ser comutado, é deixado cair.

Seguindo a proposta apresentada em [80] são considerados dois cenários de simulação. Um cenário é baseado na tecnologia actual de OXCs e respectivo hardware para processamento das mensagens de *setup* e o outro é uma projecção para um futuro próximo, considerando os próximos três a cinco anos. Para a tecnologia existente actualmente, utilizando os comutadores MEMS (*micro-electro-mechanical systems*) [163], o tempo necessário para configurar um OXC é T_{OXC}=10ms e o tempo para processar a mensagem de *setup* usando controladores JITPAC [141] é $T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT^+)$=12.5μs. Para o JET e o Horizon, tendo

em conta a literatura e o melhor conhecimento existente, estes protocolos não foram implementados em hardware. Deste modo, os autores de [80] estimam o valor do tempo de processamento da mensagem de *setup* para o JET em quatro vezes o valor do $T_{Setup}(JIT)$ e para o Horizon em duas vezes o valor do $T_{Setup}(JIT)$. Assim, $T_{Setup}(JET)=4*T_{Setup}(JIT)=50\mu s$ and $T_{Setup}(Horizon)=2*T_{Setup}(JIT)=25\mu s$.

Para o cenário do futuro próximo, estes autores apresentam uma projecção para o tempo de configuração do OXC tendo em conta uma redução em três ordens de grandeza ($T_{OXC}=20\mu s$), e para o tempo de processamento da mensagem de *setup* a redução em uma ordem de grandeza ($T_{Setup}(JET)=4\mu s$; $T_{Setup}(Horizon)=2\mu s$; $T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT^{+})=1\mu s$). Estes valores são propostos assumindo que a menor maturidade da tecnologia dos OXCs vai melhorar mais rapidamente que a tecnologia mais madura do hardware de processamento [80]. A variação destes valores entre a tecnologia actual e a do cenário do futuro próximo é uma ajuda para estudar o efeito destes tempos (T_{OXC} e T_{Setup}) na avaliação do desempenho das redes OBS.

Assume-se que a distribuição do tempo médio de duração do *burst* é $1/\mu=5*T_{OXC}=50ms$ e a taxa de chegada das mensagens de *setup* λ por nó fronteira é tal que $\lambda/\mu=32$. No entanto, em diversas abordagens efectuadas é estudada a variação do valor de λ/μ.

Em seguida são descritas as métricas para avaliação do desempenho utilizadas no estudo e depois é exposto um conjunto de figuras que procuram ilustrar e apresentar o estudo efectuado.

Métricas para Avaliação do Desempenho

Neste ponto descrevem-se três métricas utilizadas para avaliar o desempenho das redes OBS, que são as seguintes: a probabilidade de perda de *bursts*, o ganho do grau nodal e o ganho do comprimento da corda. A probabilidade de perda de *bursts* é uma métrica muito importante para a avaliação do desempenho de redes OBS [60, 66], tal como foi referido anteriormente. A probabilidade de perda de *bursts* define-se como a probabilidade da transmissão de um *burst* não atingir o seu destino. Nas redes OBS, durante a transmissão de um *burst* ao longo do seu caminho desde o nó origem até ao nó destino, este pode-se perder devido a inúmeras razões. A

probabilidade de perda de *bursts* permite medir a probabilidade dos *bursts* não atingirem o seu destino. Algumas razões podem ser invocadas para a perda de *bursts*, nomeadamente, se o pacote de controlo (mensagem de *setup*) não reservou os recursos em algum nó intermédio de interligação, o *burst* correspondente perde-se, tal como se o próprio canal de controlo sofre congestão ou outra falha que impede a sua transmissão.

A fim de quantificar os benefícios em termos do incremento do grau nodal e quantificar os benefícios que advêm da escolha do melhor comprimento da corda, são propostas duas novas métricas para a avaliação de desempenho: o ganho do grau nodal - $G_{n,k}(i,j)$, e o ganho do comprimento da corda - $G_{cl}(i,j;w_3,w_3^*)$.

O ganho do grau nodal, $G_{n,k}(i,j)$, é definido como

$$G_{n,k}(i,j) = \frac{P_i(n)}{P_j(k)} \qquad \text{(EA.2)}$$

onde $P_i(n)$ é a probabilidade de perda de *bursts* no *i*-ésimo salto (*hop*) de uma topologia de grau *n* ($P_i(n)= P_i(\text{DnT}(w_1, w_2, ..., w_n))$) e $P_j(k)$ é a probabilidade de perda de *bursts* no *j*-ésimo salto de uma topologia de grau *k*, para as mesmas condições de rede (mesmo número de canais por ligação - *F*, mesmo número de nós - *N*, mesma função de geração da distribuição, mesmo λ/μ, mesmo tempo de processamento da mensagem de *setup* - T_{Setup}, mesmo tempo de configuração do OXC - T_{OXC}, mesmo número de nós fronteira por nó de interligação, mesmo tempo de atraso entre o nó fronteira e o nó de interligação, e mesmo tempo de propagação entre nós de interligação), e para o mesmo protocolo de reserva de recursos. Na equação EA.2 utiliza-se $G_{n,k}$ onde *n* representa o grau nodal da topologia para a qual a probabilidade de perda de *bursts* no *i*-ésimo salto é $P_i(n)$ e *k* representa o grau nodal da topologia para a qual a probabilidade de perda de *bursts* no *i*-ésimo salto é $P_j(k)$.

Para quantificar os benefícios em termos do melhor comprimento da corda w_3^* em vez de w_n em redes em anel com cordas com $\text{DnT}(w_1, w_2, \ldots, w_n)$, foi introduzido o ganho do comprimento da corda, $G_{cl}(i,j; w_1, \ldots, w_n^*)$ definido como:

$$G_{cl}(i,j;w_n,w_n^*) = \frac{P_i(DnT(w_1,w_2,\ldots,w_n))}{P_j(DnT(w_1,w_2,\ldots,w_n^*))} \qquad \text{(EA.3)}$$

onde $P_i(\text{DnT}(w_1, w_2, \ldots, w_n))$ é a probabilidade de perda de *bursts* no *i*-ésimo salto de $\text{DnT}(w_1, w_2, \ldots, w_n)$, e $P_j(\text{DnT}(w_1, w_2, \ldots, w_n^*))$ é a probabilidade de perda de

bursts no *j*-ésimo salto de DnT(w_1, w_2, …, w_n^*), para as mesmas condições de rede e para o mesmo protocolo de reserva de recursos. Tendo em conta que é necessário fixar um valor para w_3, por uma questão de simplicidade, sem perda de generalidade e independente do número do nós, foi escolhido o primeiro valor não-degenerativo para w_3, isto é, w_3=3 para avaliação de $G_{cl}(i,j;\ 3,w_3^*)$ onde *i* e *j* representam o último salto de cada topologia.

Avaliação do Desempenho de Redes OBS em Anel e Anel com Cordas

Esta secção centra-se na avaliação do desempenho de redes OBS com topologias em anel e anel com cordas de grau três. Esta avaliação de desempenho baseia-se na probabilidade de perda de *bursts* obtida por simulação, utilizando a ferramenta desenvolvida no âmbito desta tese, o OBSim.

Em redes em anel com cordas, diferentes comprimentos de corda podem conduzir a diferentes diâmetros da rede, e assim, a um número máximo de saltos diferente em cada rede. Um resultado interessante encontrado prende-se com os diâmetros da família D3T(w_1,w_2,w_3), para a qual w_2=(w_1+2) *mod N* ou w_2=(w_1-2) *mod N*. Cada família deste tipo, isto é, D3T(w_1, (w_1+2) *mod N*, w_3) ou D3T(w_1, (w_1-2) *mod N*, w_3), com $1 \leq w_1 \leq 19$ e $w_1 \neq w_2 \neq w_3$, tem um diâmetro que é uma versão deslocada (em relação a w_3) do diâmetro da família de anéis com cordas (D3T(1,*N*-1,w_3)). Por esta razão, a análise centra-se em redes de anéis com cordas, ou seja, nas topologias D3T(1,19,w_3), D4T(1,19,3,w_4) e D3T(1,19,5,w_4). A Figura V apresenta os diâmetros de D3T(1,3,*w*3), D3T(1,19,*w*3), D3T(3,5,*w*3) e D3T(5,7,*w*3), que ilustram esta situação. De notar que o diâmetro de D3T(3,1,*w*3) é o mesmo diâmetro de D3T(1,3,*w*3) e que o diâmetro de D3T(3,5,*w*3) é o mesmo diâmetro de D3T(5,3,*w*3), isto é, em anéis com cordas, a ordem do comprimento da corda é comutativa.

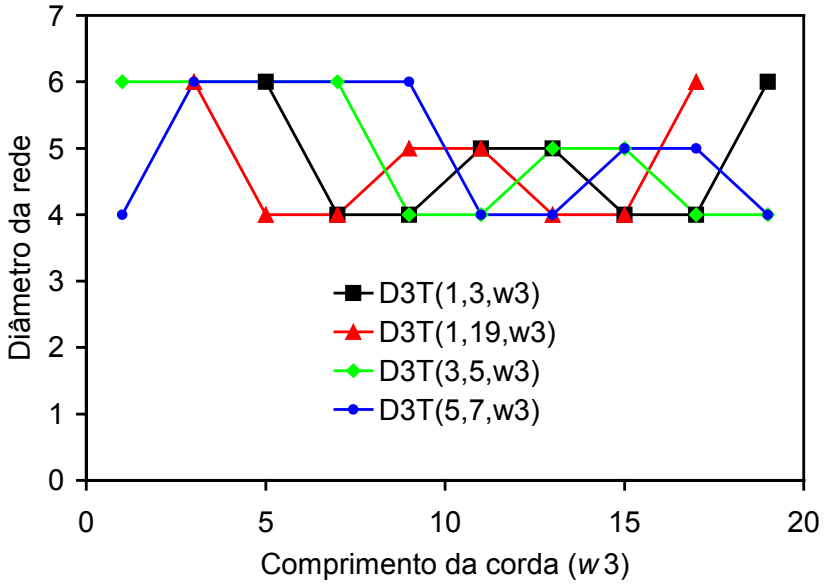


Fig. V. Diâmetro da rede em função do comprimento da corda para D3T(1,3,w_3), D3T(1,19,w_3), D3T(3,5,w_3) e D3T(5,7,w_3), com $1 \leq w3 \leq 19$ e $w1 \neq w2 \neq w3$.

A Figura VI apresenta a probabilidade de perda de *bursts*, com N=20 nós, em função do comprimento da corda para topologias de grau três (D3T(1,19,w_3)) e grau quatro (D4T(1,19,3,w_4) and (D4T(1,19,5,w_4)), para F=64 canais de dados por ligação. Como se pode observar, para D3T(1,19,w_3), o melhor desempenho da rede é obtido para w_3=5 e w_3=7, que corresponde ao diâmetro mínimo da rede com valor 4 para esta topologia. Assim, D3T(1,19,7) é escolhida como a topologia em anel com cordas de grau três com o melhor desempenho. Para D4T(1,19,3,w_4), o melhor desempenho da rede é obtido para w_4=9 e w_4=13, que corresponde ao diâmetro mínimo de rede com valor 3 para esta topologia. Para D4T(1,19,5,w_4), o melhor desempenho da rede é obtido para w_4=9 e w_4=11, que corresponde ao diâmetro mínimo de rede com valor 3 para esta topologia. Como se pode verificar, para topologias de grau quatro, o comprimento de corda w_4=9 tem um desempenho ligeiramente superior para ambas as topologias. Deste modo, as topologias D4T(1,19,3,9) e (D4T(1,19,5,9) são escolhidas como topologias em anel com cordas de grau quatro com o melhor desempenho, com um comportamento ligeiramente superior para D4T(1,19,3,9).

A Figura VII mostra a probabilidade de perda de *bursts* no último salto de D2T(1,19), D3T(1,19,7) e D4T(1,19,3,9), para os protocolos de reserva de recursos JIT, JumpStart, JIT$^+$, JET e Horizon. Como se pode observar nesta figura, os anéis com

cordas têm claramente melhor desempenho que os anéis e, em termos de anéis com cordas, as topologias de grau quatro têm melhor desempenho que as de grau três.

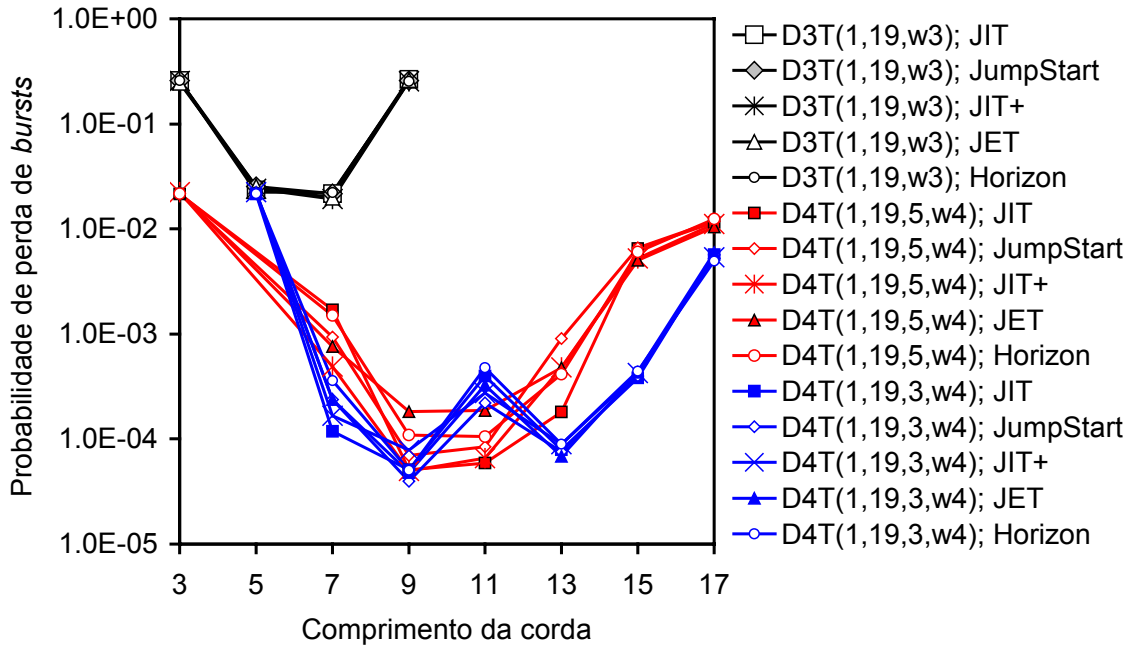


Fig. VI. Probabilidade de perda de *bursts versus* comprimeto da corda para D3T(1,19,w_3), D4T(1,19,3,w_4) e D4T(1,19,5,w_4); N=20; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

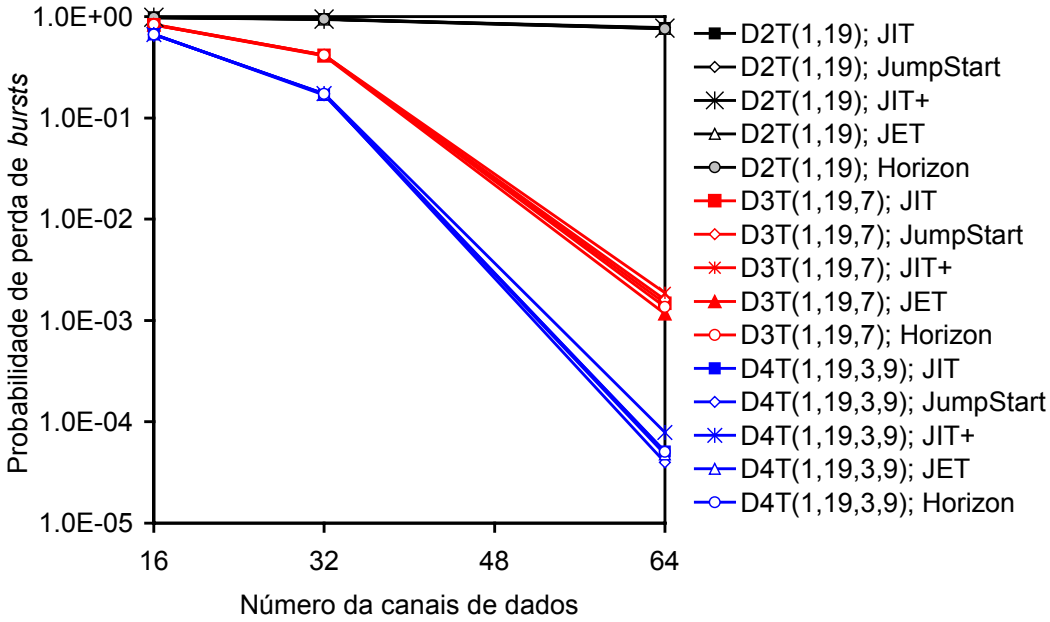


Fig. VII. Probabilidade de perda de *bursts* no último salto de cada topologia *versus* número de canais de dados para D2T(1,19), D3T(1,19,7), and D4T(1,19,3,9); N=20; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

A Figura VIII ilustra a probabilidade de perda de *bursts*, para *N*=20 nós, para D2T(1,19), D3T(1,19,7) e D4T(1,19,3,9) em função do número de saltos para o JIT, JumpStart, JIT$^+$, JET e Horizon. Esta figura confirma claramente os resultados da Figura VII, isto é, os anéis com cordas de grau quatro apresentam melhor desempenho que os anéis e as redes em anel com cordas de grau três.

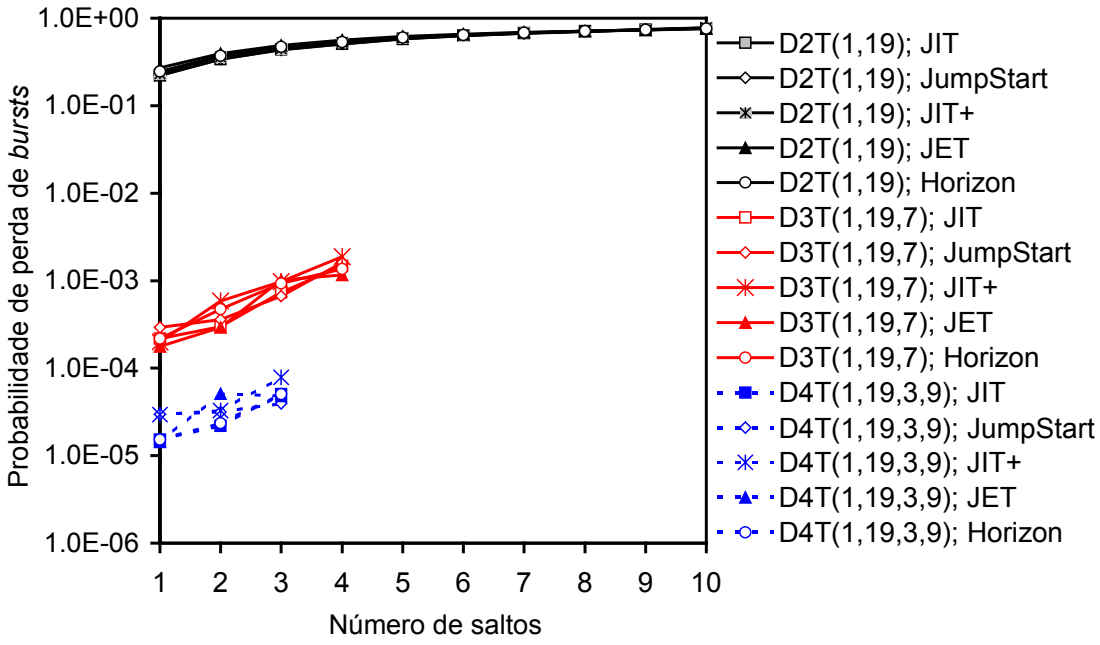


Fig. VIII. Probabilidade de perda de *bursts versus* número de saltos para D2T(1,19), D3T(1,19,7) e D4T(1,19,3,9); *N*=20; *F*=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

A próxima figura centra-se no ganho do grau nodal descrito anteriormente. Assim, a Figura IX apresenta o ganho do grau nodal, $G_{n,k}(i,j)$ no último salto de cada topologia, dado o incremento do grau nodal de 2 (D2T(1,19)) para 3 (D3T(1,19,7)), e de 2 (D2T(1,19)) para 4 (D4T(1,19,3,9)) para os protocolos JIT, JumpStart, JIT$^+$, JET e Horizon (*F*=64). Para *F*=16 e *F*=32, o ganho do grau nodal é muito pequeno dada a alta probabilidade de perda de *bursts*. No entanto, quando o número de canais de dados por ligação aumenta para 64, observa-se um ganho do grau nodal entre uma e três ordens de grandeza para anéis com cordas de grau três. Além disso, o aumento do grau nodal de 2 (D2T(1,19)) para 4 (D4T(1,19,3,9)) conduz a uma melhoria de desempenho entre 3 e 4 ordens de grandeza. Outra observação importante que pode ser feita a partir desta figura é que os cinco protocolos de reserva de recursos em

estudo apresentam ganhos do grau nodal semelhantes, quer quando o grau nodal aumenta de 2 para 3 ($G_{2,3}(10,4)$) quer quando o grau nodal aumenta de 2 para 4 ($G_{2,4}(10,3)$).

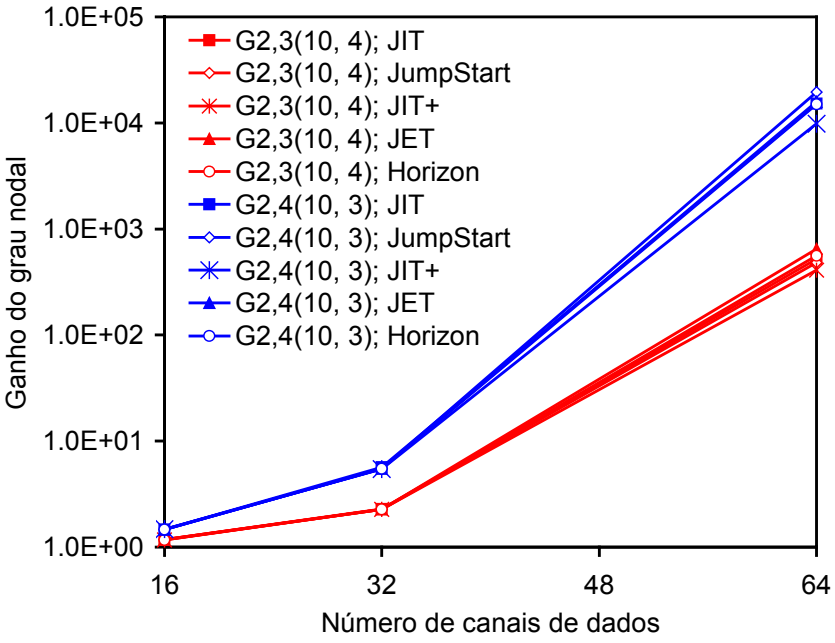


Fig. IX. Ganho do grau nodal, no último salto da cada topologia, dado o incremento do grau nodal de 2 (D2T(1,19)) para 3 (D3T(1,19,7)) e para 4 D4T(1,19,3,9); N=20; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

As próximas figuras focam a influência do ganho do comprimento da corda. A Figura X mostra o ganho do comprimento da corda, G_{cl} (6,4; 3,w_3^*), como função do número de canais de dados, no último salto de cada D3T(1,19,w_3^*) para w_3^*=5 e w_3^*=7. Como se pode observar, o ganho do comprimento da corda no último salto de cada topologia é muito pequeno para F=16 ou F=32 canais de dados por ligação, dada a elevada probabilidade de perda de *bursts* para esse número de canais de dados. É possível verificar que o ganho do comprimento da corda é idêntico. Para F=64 canais de dados, o maior ganho do comprimento da corda no último salto é obtido para w_3^*=7 e é ligeiramente menor que duas ordens de grandeza.

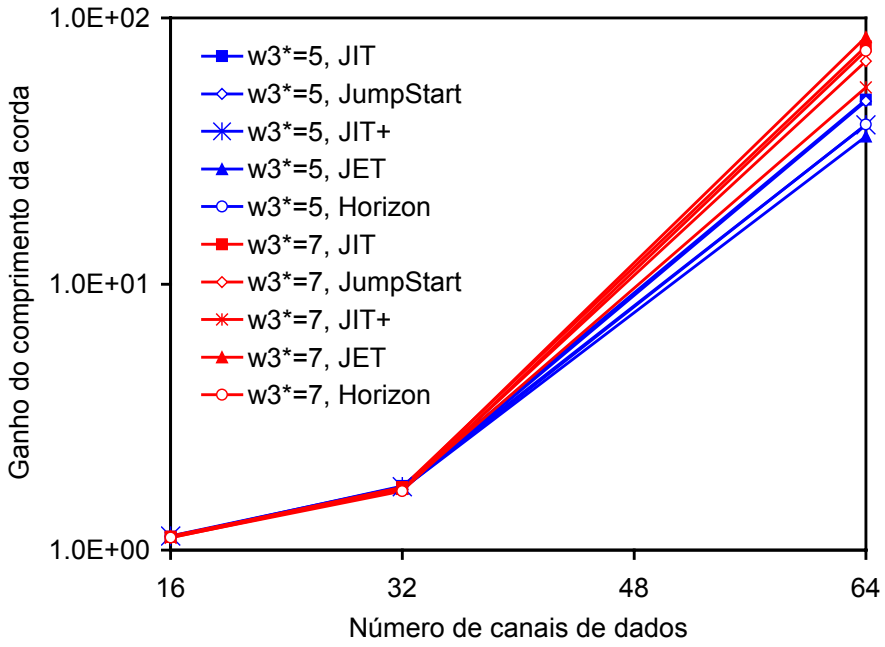


Fig. X. Ganho do comprimento da corda G_{cl} (6,4; 3,w_3^*), em função do número de canais de dados, no último salto de D3T(1,19,w_3) para a escolha de w_3^*=5 ou w_3^*=7, em vez de w_3=3; F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

A Figura XI apresenta os ganhos do comprimento da corda G_{cl}(6,j; 3,w_3^*) no último salto de cada D3T(1,19,w_3), em função do comprimento da corda para o JIT, JumpStart, JIT$^+$, JET e Horizon. Esta figura mostra claramente que os comprimentos de corda de w_3=5 e w_3=7 conduzem a melhores desempenhos, sendo o desempenho de w_3=7 ligeiramente melhor que o desempenho para w_3=5. O ganho destes comprimentos de corda situa-se entre uma e duas ordens de grandeza, o que confirma as observações anteriores.

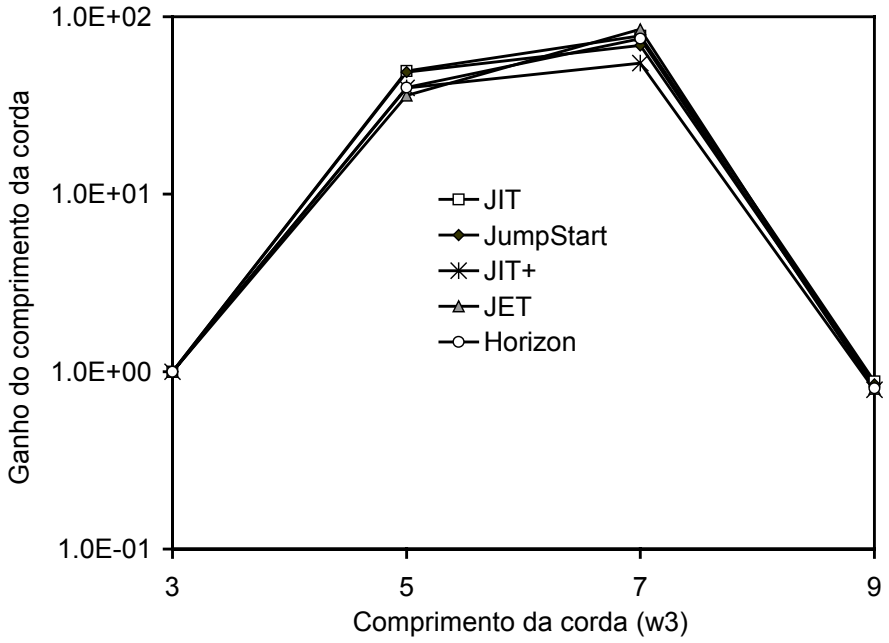


Fig. XI. Ganho do comprimento da corda $G_{cl}(6,j;\ 3,w_3^*)$ em função do comprimento da corda, no último salto de cada D3T(1,19,w_3^*) para o JIT, JumpStart, JIT$^+$, JET e Horizon; F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

As próximas figuras centram-se na influência dos tempos de processamento da mensagem de *setup* e de configuração do OXC no desempenho de redes em anel com cordas de grau três e grau quatro para os protocolos de reserva de recursos JIT, JumpStart, JIT$^+$, JET e Horizon. As redes em anel com cordas de grau três e grau quatro que foram escolhidas são D3T(1,19,7) e D4T(1,19,3,9), respectivamente, pois são as que apresentam melhor desempenho. Em termos das redes em anel, não exibidas nestas figuras, estas apresentam um desempenho muito inferior ao anel com cordas de grau três, sobretudo para valores de $T_{OXC}\leq$1ms, que se cifra num valor de quatro ordens de grandeza menor.

A Figura XII mostra a probabilidade de perda de *bursts* em função do tempo de configuração do OXC, no último salto de D3T(1,19,7) e D4T(1,19,3,9) para F=64 e λ/μ=44.8. Nesta figura o tempo T_{Setup} é definido para o JIT, JumpStart e JIT$^+$ e é estimado para o JET e Horizon tendo em conta a actual tecnologia disponível (os controladores JITPAC [141]). Para o tempo T_{OXC} é assumida uma gama de valores entre o valor estimado para um cenário de futuro próximo (T_{OXC} =20μs) e dez vezes

o valor definido para a tecnologia actual disponível, isto é, T_{OXC}=10*10ms=100ms. Como se pode observar na figura, a topologia em anel com cordas de grau quatro apresenta claramente um melhor desempenho que as topologias de grau três. Também se pode observar que para $T_{OXC}\leq$1ms, o desempenho dos diferentes protocolos apresenta algumas oscilações. Apesar disso, é possível concluir que apesar da melhoria e do desenvolvimento de novas tecnologias, o desempenho da rede já não apresenta melhorias e o T_{OXC} (para valores inferiores a 1ms) não influencia o desempenho dessas redes. No entanto, quando $T_{OXC}>$1ms, é possível observar que o valor da probabilidade de perda de *bursts* aumenta com o incremento do valor de T_{OXC} entre 1ms e 100ms. Este comportamento expressa a quantidade de tempo em que os recursos do OXC estão reservados para um *burst*. Os resultados apresentados na Figura XII são confirmados pelos da Figura XIII. Nesta figura, assume-se que a variação do T_{Setup} é efectuada em função da variação do T_{OXC} de acordo com uma interpolação linear. O T_{OXC} assume valores que vão desde 20μs até aos 100ms, considerando os seguintes tempos intermédios: 0.1ms, 1ms, 10ms e 50ms. Deste modo, o valor do T_{Setup} para os protocolos JIT, JumpStart e JIT$^+$, onde *X* representa o protocolo de reserva de recursos, é dado por:

$$T_{Setup}(X) = 1 + \frac{11.5}{10^4 - 20}(T_{OXC}(X) - 20) \quad (\mu s) \tag{EA.4}$$

O T_{Setup} para o protocolo JET é dado por:

$$T_{Setup}(JET) = 4 + \frac{46}{10^4 - 20}(T_{OXC}(JET) - 20) \quad (\mu s) \tag{EA.5}$$

Para o protocolo de reserva de recursos Horizon T_{Setup} é dado por:

$$T_{Setup}(Horizon) = 2 + \frac{23}{10^4 - 20}(T_{OXC}(Horizon) - 20) \quad (\mu s) \tag{EA.6}$$

A Figura XIII ilustra a probabilidade de perda de *bursts* em função do tempo de configuração do OXC no último salto de D3T(1,19,7) e D4T(1,19,3,9) para os cinco protocolos em estudo, com *F*=64 e λ/μ=44.8. Assume-se que o T_{Setup} varia em função do T_{OXC}, de acordo com (EA.4), (EA.5) e (EA.6). Como se pode observar, para valores de T_{OXC} menores que 1ms não existe um impacto significativo no desempenho das redes. Os resultados desta figura também confirmam os da figura anterior acerca da influência do T_{Setup} e do T_{OXC} no desempenho da rede.

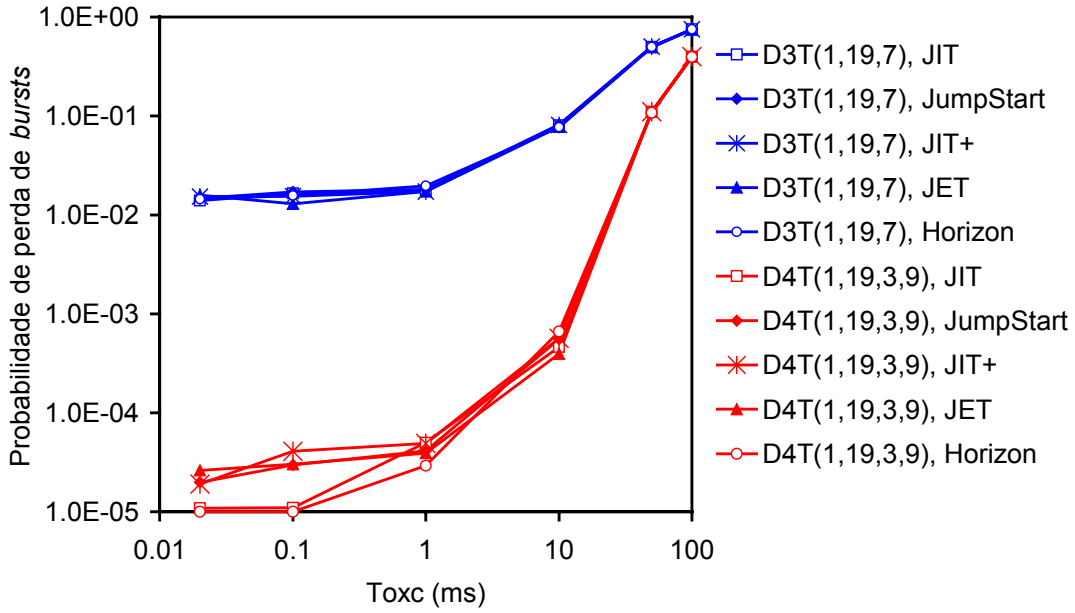


Fig. XII. Probabilidade de perda de *bursts* em função do tempo de configuração do OXC no último salto de D3T(1,19,7) e D4T(1,19,3,9) para o JIT, JumpStart, JIT$^+$, JET e Horizon; F=64; λ/μ=44.8; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)= 50μs; T_{Setup}(Horizon)=25μs.

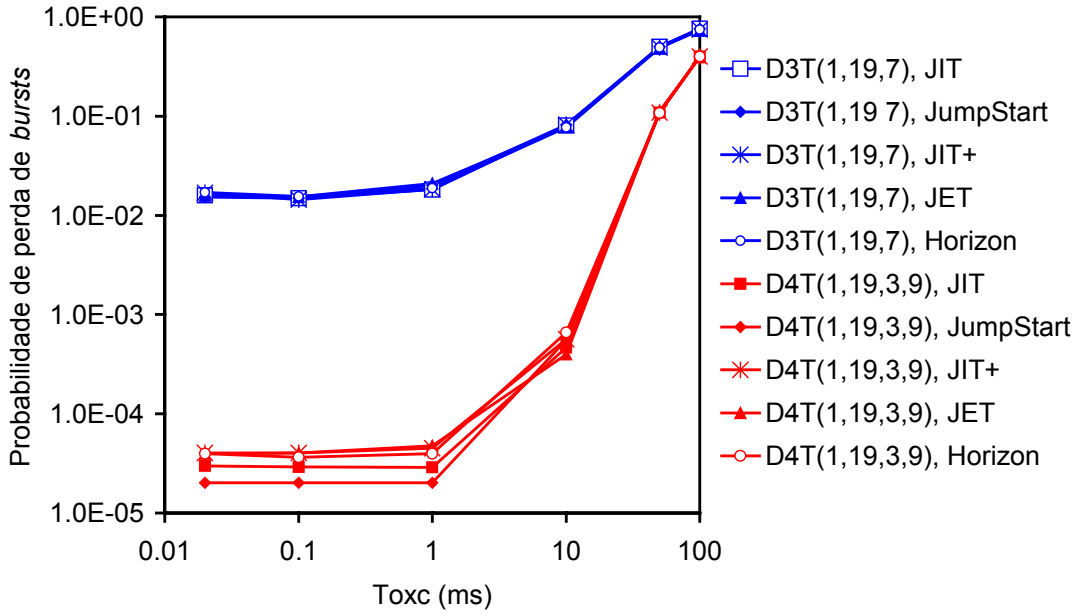


Fig. XIII. Probabilidade de perda de *bursts* em função do tempo de configuração do OXC no último salto de D3T(1,19,7) e D4T(1,19,3,9) para o JIT, JumpStart, JIT$^+$, JET e Horizon; F=64; λ/μ=44.8; com a variação de T_{Setup} de acordo com (EA.4), (EA.5) e (EA.6) para cada protocolo.

A Figura XIV apresenta a probabilidade de perda de *bursts* em função do tempo de processamento da mensagem de *setup* (T_{Setup}) no último salto de D3T(1,19,7) e D4T(1,19,3,9) para o JIT, JumpStart, JIT$^+$, JET e Horizon, com F=64 e λ/μ=44.8. Em relação ao valor de T_{OXC} são considerados dois cenários, um assumindo o valor para a actual tecnologia existente (com T_{OXC}=10ms) e outro estimando o valor para um cenário de futuro próximo (T_{OXC}=20μs). Para cada curva da Figura XIV, é assumido um valor fixo para T_{OXC} enquanto o T_{Setup} varia entre os valores considerados para a tecnologia actual existente e os valores estimados para um futuro próximo. Assim, o T_{Setup} varia entre 1μs e 12.5μs, para o JIT, JumpStart e JIT$^+$, entre 2μs e 25μs para o JET, e entre 4μs e 50μs para o Horizon. Como se pode observar, o desempenho do anel com cordas de grau quatro é claramente melhor que o de grau três e o comportamento dos cinco protocolos é muito semelhante. Esta figura mostra que com a redução do T_{Setup}, o desempenho da topologia de grau quatro é ligeiramente melhor, principalmente, para T_{OXC}=20μs. Por outro lado, com a redução do tempo de configuração do OXC obtém-se um desempenho melhor em ambas as redes. Além disso, a influência do grau nodal no desempenho de uma dada topologia é maior que a influência do T_{OXC}, como se pode observar entre as topologias de rede de grau três e grau quatro. Também se pode observar que para os anéis com cordas de grau quatro, uma redução do T_{OXC} de 10ms até 20μs conduz a uma melhoria de desempenho a rondar três ordens de grandeza. Para anéis com cordas de grau três, a redução do T_{OXC} leva a uma melhoria de desempenho à volta de duas ordens de grandeza.

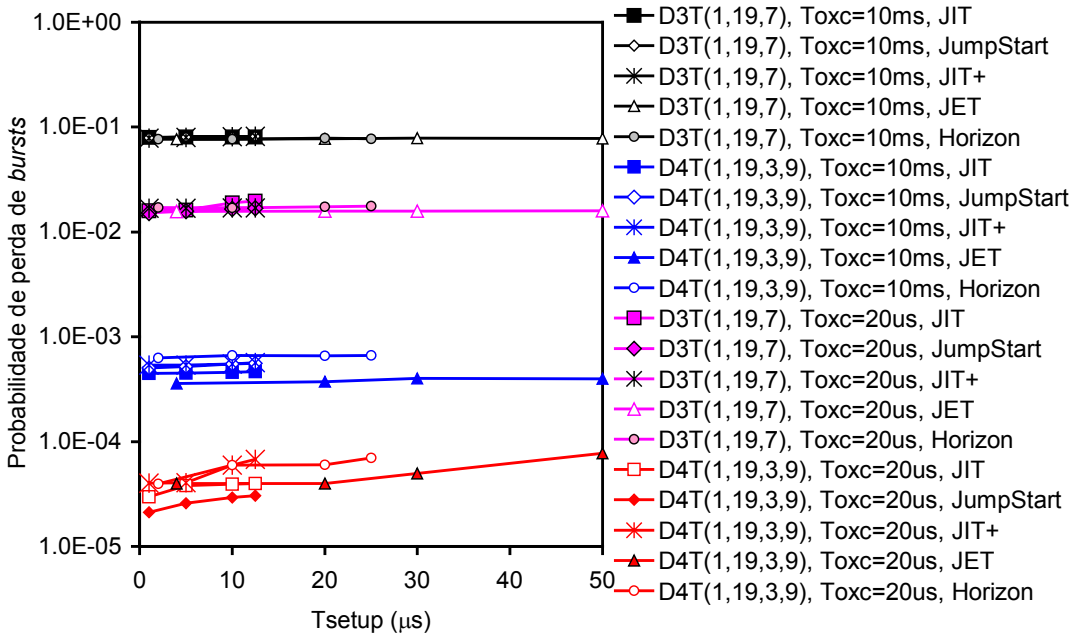


Fig. XIV. Probabilidade de perda de *bursts* em função do tempo de processamento da mensagem de *setup* no último salto de D3T(1,19,7) e D4T(1,19,3,9) para o JIT, JumpStart, JIT$^+$, JET e Horizon; F=64; λ/μ=32; T_{OXC}=10ms; T_{OXC}=20μs.

A Figura XV assume a variação do T_{OXC} em função da variação do T_{Setup} resolvendo as equações (EA.4), (EA.5) e (EA.6) em ordem a T_{OXC}. T_{Setup} assume valores que variam entre 1μs e 12.5μs, considerando 5μs e 10μs como tempos intermédios para os protocolos JIT, JumpStart e JIT$^+$. Deste modo, o valor de T_{OXC} para estes protocolos, onde X corresponde ao respectivo protocolo de reserva de recursos, é dada por:

$$T_{OXC}(x) = 20 + \frac{(T_{Setup}(x) - 1)(10^4 - 20)}{11.5} \quad (\mu s) \qquad \text{(EA.7)}$$

O T_{Setup} para o JET assume valores que variam entre 4μs e 50μs, considerando 20μs e 30μs como tempos intermédios. T_{OXC} é dado por:

$$T_{OXC}(JET) = 20 + \frac{(T_{Setup}(JET) - 4)(10^4 - 20)}{46} \quad (\mu s) \qquad \text{(EA.8)}$$

O T_{Setup} para o Horizon assume valores que variam entre 2μs e 25μs, considerando os tempos 10μs e 20μs como valores intermédios. T_{OXC} é dado por:

$$T_{OXC}(Horizon) = 20 + \frac{(T_{Setup}(Horizon) - 2)(10^4 - 20)}{23} \quad (\mu s) \qquad \text{(EA.9)}$$

Como seria de esperar, os valores de T_{OXC} calculados de acordo com (EA.7), (EA.8) e (EA.9) para cada protocolo são muito semelhantes. Deste modo, a Figura XV considera a probabilidade de perda de *bursts* em função do tempo de processamento da mensagem de *setup* no último salto de D3T(1,19,7) e D4T(1,19,3,9) para os protocolos JIT, JumpStart, JIT$^+$, JET e Horizon, sendo o T_{OXC} calculado de acordo com (EA.7), (EA.8) e (EA.9) para cada protocolo. Como se pode observar, quando o valor de T_{Setup} aumenta, o correspondente valor de probabilidade de *bursts* também aumenta, sendo à volta de uma ordem de grandeza para o anel com cordas de grau quatro. Esta figura confirma os resultados anteriores no que diz respeito ao melhor desempenho do anel com cordas de grau quatro em comparação com o de grau três, variando entre duas e três ordens de grandeza. Este resultado também confirma a influência do T_{OXC} e do T_{Setup} no desempenho de redes OBS.

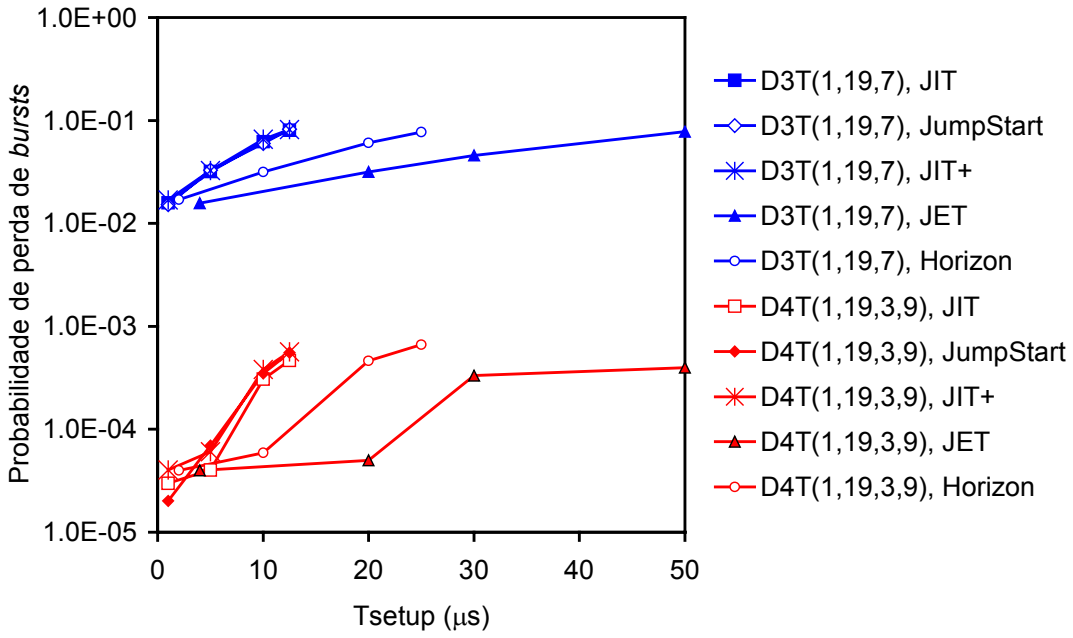


Fig. XV. Probabilidade de perda de *bursts versus* T_{Setup} no último salto de D3T(1,19,7) e D4T(1,19,3,9) para o JIT, JumpStart, JIT$^+$, JET e Horizon; F=64; λ/μ=32; com a variação de T_{OXC} de acordo com (EA.7), (EA.8) e (EA.9) para cada protocolo.

Avaliação do Desempenho de Redes OBS com topologias em Malha

Esta secção centra-se na avaliação do desempenho de redes OBS com topologias em malha para os protocolos de reserva de recursos JIT, JumpStart, JIT$^+$, JET e Horizon. Este estudo baseia-se na probabilidade de perda de *bursts* obtida por simulação, recorrendo à ferramenta descrita anteriormente, o OBSim. Os parâmetros de simulação utilizados são os mesmos que foram descritos na secção anterior.

A análise centra-se em redes OBS com topologias em malha apresentadas anteriormente e que são as seguintes: anéis com cordas com o número de nós a variar entre 10 e 30, teia toroidal com 16 e 25 nós, a NSFNET com 14 nós e 21 ligações, a NSFNET com 16 nós e 25 ligações, a ARPANET com 20 nós e 32 ligações, a rede óptica europeia (*European Optical Network* - EON) com 19 nós e 37 ligações e a rede proposta para a FCCN (FCCN-NET) com 14 nós e 14 ligações. Para efeitos, comparativos as topologias em anel bidireccional também são consideradas. Estas topologias têm os seguintes graus nodais: anel, 2.0; anel com cordas de grau três, 3.0; anel com cordas de grau quatro, 4.0; anel com cordas de grau cinco, 5.0; anel com cordas de grau seis, 6.0; teia toroidal, 4.0; NSFNET com 14 nós e 21 ligações, 3.0; NSFNET com 16 nós e 25 ligações, 3.125; ARPANET com 20 nós e 32 ligações, 3.2; EON com 19 nós e 37 ligações, 3.89; e a FCCN-NET com 14 nós e 14 ligações, 2.0. O grau nodal é a média do número de ligações entre nós de uma determinada topologia e é calculada em função do número de nós e do número de ligações. Por exemplo, considerando uma rede com N nós e L ligações, como cada ligação é bidireccional, o número total de ligações é de $2L$. Então, dividindo o número de ligações ($2L$) unidireccionais pelo número de nós (N), obtém-se o valor do grau nodal igual a $2L/N$.

Em seguida, apresentam-se algumas figuras que procuram ilustrar o desempenho de redes OBS com topologias em malha. A Figura XVI mostra a probabilidade de perda de *bursts* em função do número de nós (N), entre N=10 e N=30, no último salto de redes em anel, anel com cordas de grau três e grau quatro, FCCN-NET, NSFNET, ARPANET, EON e teia toroidal para o protocolo JIT. Resultados

similares foram encontrados para os protocolos Jumpstart, JIT$^+$, JET e Horizon, confirmando resultados anteriores para redes em anel e anel com cordas. Assim, as figuras com estes protocolos não são apresentadas no presente resumo. Apesar do desempenho dos diferentes protocolos ser similar, os protocolos de reserva imediata (JIT, JIT$^+$ e JumpStart) para redes OBS em malha são mais apropriados, uma vez que a sua implementação é mais simples do que a dos protocolos de reserva atrasada. A Figura XVII mostra as diâmetros de rede correspondentes. Como se pode observar, quando o número de nós é pequeno a D3T(1,*N*-1,5) tem o melhor desempenho. Quando o número de nós é maior, isto é, com mais de 24 nós, D3T(1,*N*-1,7) tem melhor desempenho que D3T(1,*N*-1,5). Contudo, para redes cujo número de nós seja superior a 16, D4T(1,*N*-1,5,9) tem o melhor desempenho. O pior desempenho verifica-se para os anéis e para a FCCN-NET.

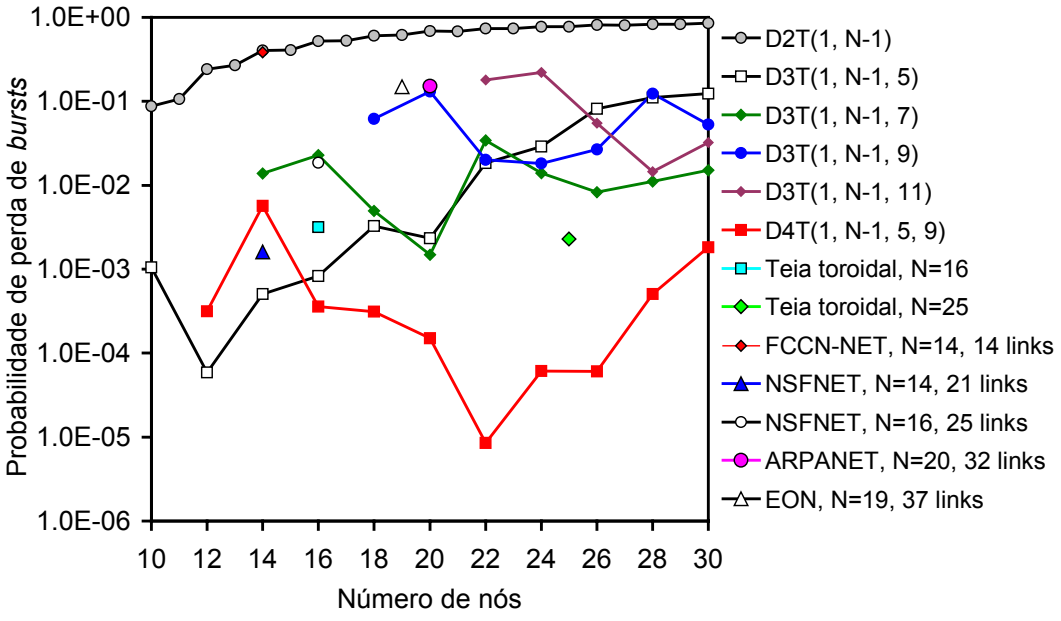


Fig. XVI. Probabilidade de perda de *bursts* em função do número de nós (*N*), no último salto de anéis, anéis com cordas de grau três e quatro, FCCN-NET, NSFNET, ARPANET, rede óptica europeia e teia toroidal para o protocolo JIT; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

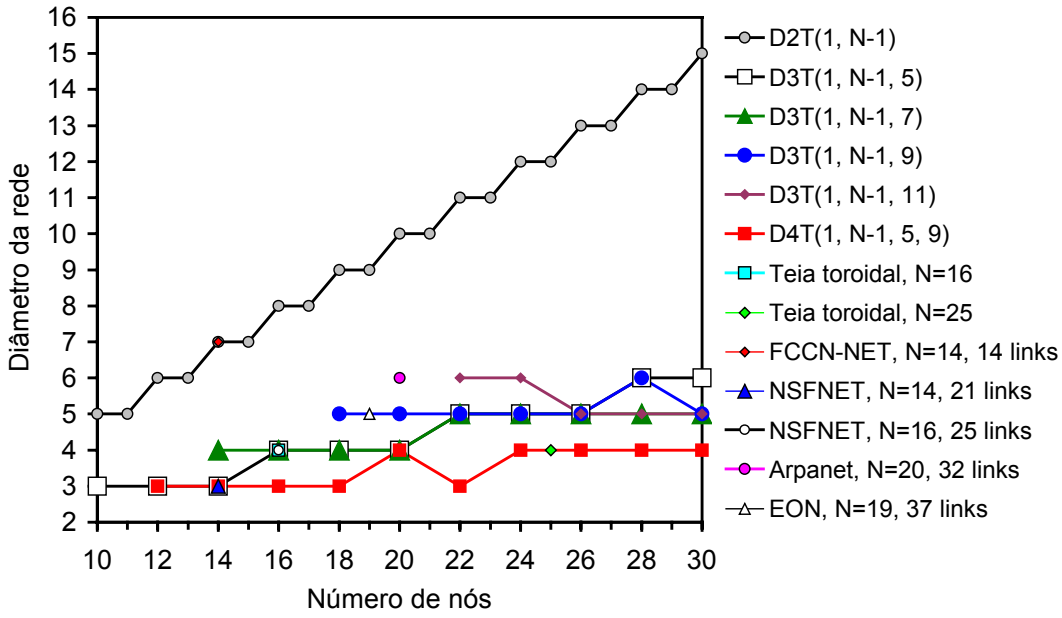


Fig. XVII. Diâmetro da rede em função do número de nós (*N*) redes em anel, anel com cordas de grau três e quatro, FCCN-NET, NSFNET, ARPANET, rede óptica europeia e teia toroidal.

As próximas figuras concentram-se na avaliação de desempenho de redes OBS com topologias em malha com 16 nós e à volta de 20 nós, respectivamente. Redes com 16 nós são a NSFNET, a teia toroidal e os anéis com cordas de grau três a seis, e redes com o número de nós a rondar os 20 nós são a ARPANET, a rede óptica europeia (com 19 nós), e os anéis com cordas de grau nodal entre três e seis. No que diz respeito aos anéis com cordas, de entre as topologias de diâmetro mínimo foram escolhidas aquelas que apresentam melhor desempenho para cada grau nodal. Esta análise considera os cinco protocolos de reserva de recursos JIT, JumpStart, JIT$^+$, JET e Horizon e centra-se no papel do ganho do grau nodal em redes OBS com topologias em malha. A FCCN-NET não pode ser considerada nestas figuras uma vez que esta topologia tem o mesmo grau nodal (igual a 2) que a topologia em anel correspondente com o mesmo número de nós (com *N*=14).

A Figura XVIII ilustra o ganho do grau nodal no último salto de cada topologia, dado o incremento de grau nodal de 2 (D2T(1,15)) para: 3 (D3T(1,15,5)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) e teia toroidal), 5 (D5T(1,15,7,3,9)) e 6 (D6T(1,15,3,5,7,11) para o protocolo JIT. Como se pode observar na figura, as topologias podem ser ordenadas desde a topologia com melhor desempenho para o pior da seguinte forma: D6T(1,15,3,5,7,11), D5T(1,15,7,3,9), D4T(1,15,5,13),

D3T(1,15,5), teia toroidal e NSFNET. De notar que o desempenho de D5T(1,15,7,3,9) é próximo do de D6T(1,15,3,5,7,11). Também foram calculados os resultados para os protocolos JumpStart, JIT$^+$, JET e Horizon, mas como são bastante semelhantes, as respectivas figuras não são apresentadas neste resumo. Para 16 nós e 64 canais de dados por ligação, quando o grau nodal aumenta de 2 (anel) para 4, 5 e 6 (anéis com cordas de diâmetro mínimo), observa-se o maior ganho entre quatro e cinco ordens de grandeza no último salto de cada topologia. No que concerne ao ganho do grau nodal, dado o aumento do grau nodal de 2 (anel) para 3 (anel com cordas de diâmetro mínimo), este cifra-se à volta de três ordens de grandeza no último salto de cada topologia para as mesmas condições de rede (N=16 e F=64). Comparando o ganho do grau nodal nos anéis com cordas, a maior diferença entre dois graus nodais consecutivos verifica-se entre os graus três e quatro, sendo mais de uma ordem de grandeza, excepto para o JIT$^+$ que é menor que uma ordem de grandeza (não exibido na figura). O desempenho dos cinco protocolos em estudo é muito semelhante.

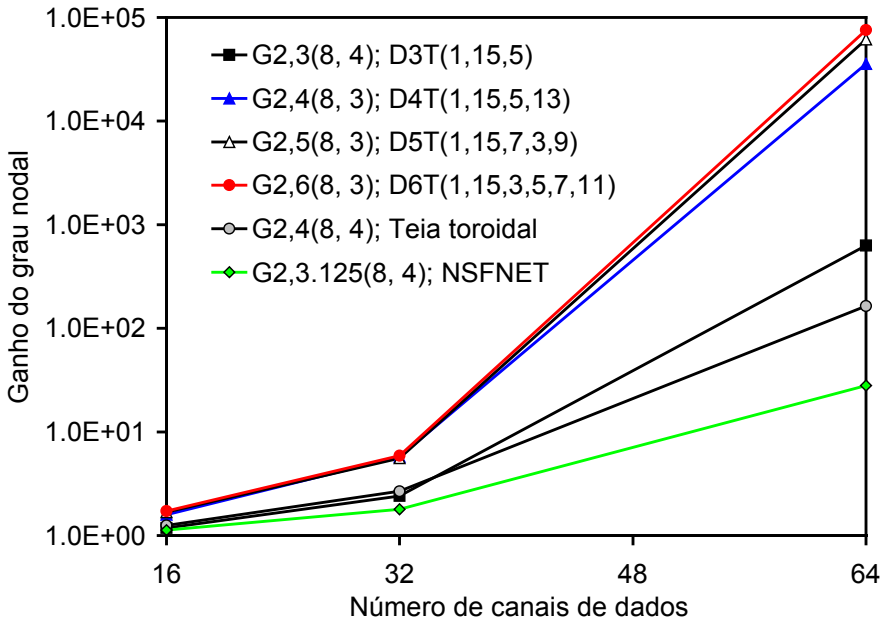


Fig. XVIII. Ganho do grau nodal dado o aumento do grau nodal de 2 (D2T(1,15)) para: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) e teia toroidal), 5 (D5T(1,15,7,3,9)) e 6 (D6T(1,15,3,5,7,11) em função do número de canais de dados, no último salto de cada topologia, para o protocolo JIT; N=16; λ/μ=32; T_{OXC}=10ms; T_{setup}(JIT)=T_{setup}(JumpStart)=T_{setup}(JIT$^+$)=12.5μs; T_{setup}(JET)=50μs; T_{setup}(Horizon)=25μs.

A Figura XIX mostra o ganho do grau nodal no último salto de cada topologia, para o protocolo JIT, dado o incremento de grau nodal de 2 (D2T(1,19)) para: 3 (D3T(1,19,7)), 3.2 (ARPANET), 4 (D4T(1,19,3,9)), 5 (D5T(1,19,3,7,11)) e 6 (D6T(1,19,3,5,11,15)), e de 2 (D2T(1,18)) para 3.89 (rede óptica europeia). Como se pode observar na figura, as topologias podem ser ordenadas desde o melhor para o pior desempenho como: D6T(1,19,3,5,11,15), D5T(1,19,3,7,11), D4T(1,19,3,9), D3T(1,19,7), ARPANET e rede óptica europeia. Para redes com 20 nós, quando o grau nodal aumenta de 2 para 4, 5 e 6 (anel com cordas), verifica-se um ganho entre as quatro e cinco ordens de grandeza. Observa-se que o desempenho da ARPANET é muito semelhante ao da rede óptica europeia. Apesar do grau nodal da ARPANET (3.2) ser próximo do anel com cordas de grau três (D3T(1,19,7)) e do grau nodal da rede óptica europeia (3.89) ser próximo do anel com cordas de grau quatro (D4T(1,19,3,9)), o desempenho dos anéis com cordas é muito melhor. No caso da D4T(1,19,3,9) o seu desempenho é cerca de quatro ordens de grandeza melhor que a rede óptica europeia. O desempenho do ganho do grau nodal desta rede e da ARPANET são piores que o anel com cordas de grau três, apresentando valores inferiores a uma ordem de grandeza. Estes resultados revelam a importância do modo como as ligações são efectuadas numa rede OBS, uma vez que alguns anéis com cordas têm um grau nodal próximo do da ARPANET e rede óptica europeia e apresentam um desempenho bastante superior. As figuras com os resultados para os restantes quatro protocolos não são apresentadas porque o seu desempenho é similar. Este resultado é confirmado na Figura XX. Esta figura apresenta uma comparação do desempenho para a melhor topologia (D6T(1,19,3,5,11,15)) e para a pior (rede óptica europeia - EON) para redes com o número de nós à volta de 20. A Figura XX mostra o ganho do grau nodal em função de λ/μ, incrementando o grau nodal desde 2 (D2T(1,18)) até 3.89 (rede óptica europeia), e desde 2 (D2T(1,19)) até 6 (D6T(1,19,3,5,11,15)), no último salto de cada topologia para os protocolos JIT, JumpStart, JIT$^+$, JET e Horizon (F=64).

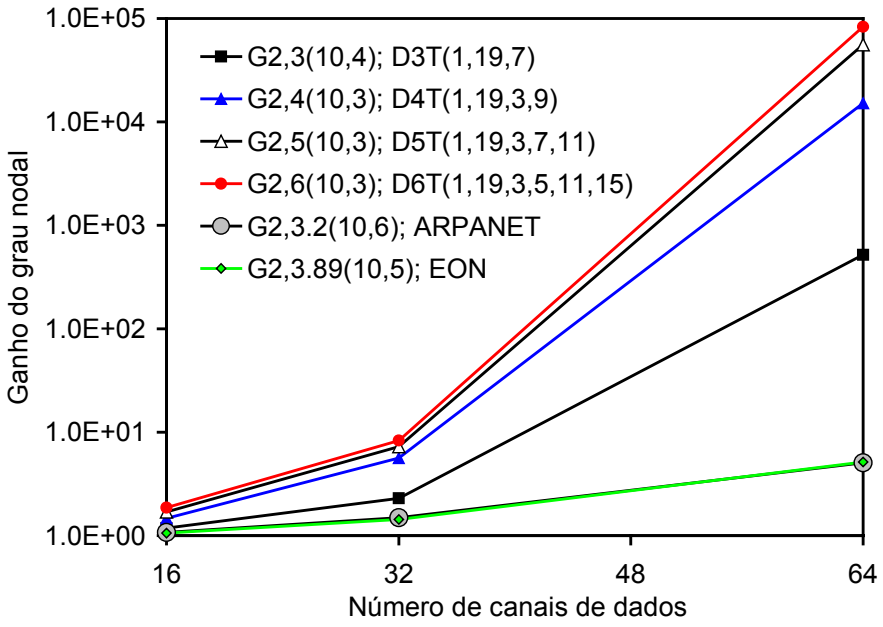


Fig. XIX. Ganho do grau nodal dado o aumento do grau nodal de 2 (D2T(1,19)) para: 3 (D3T(1,19,7)), 3.2 (ARPANET), 3.89 (EON), 4 (D4T(1,19,3,9)), 5 (D5T(1,19,3,7,11)) e 6 (D6T(1,19,3,5,11,15)) em função do número de canais de dados, no último salto de cada topologia, para o protocolo JIT; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

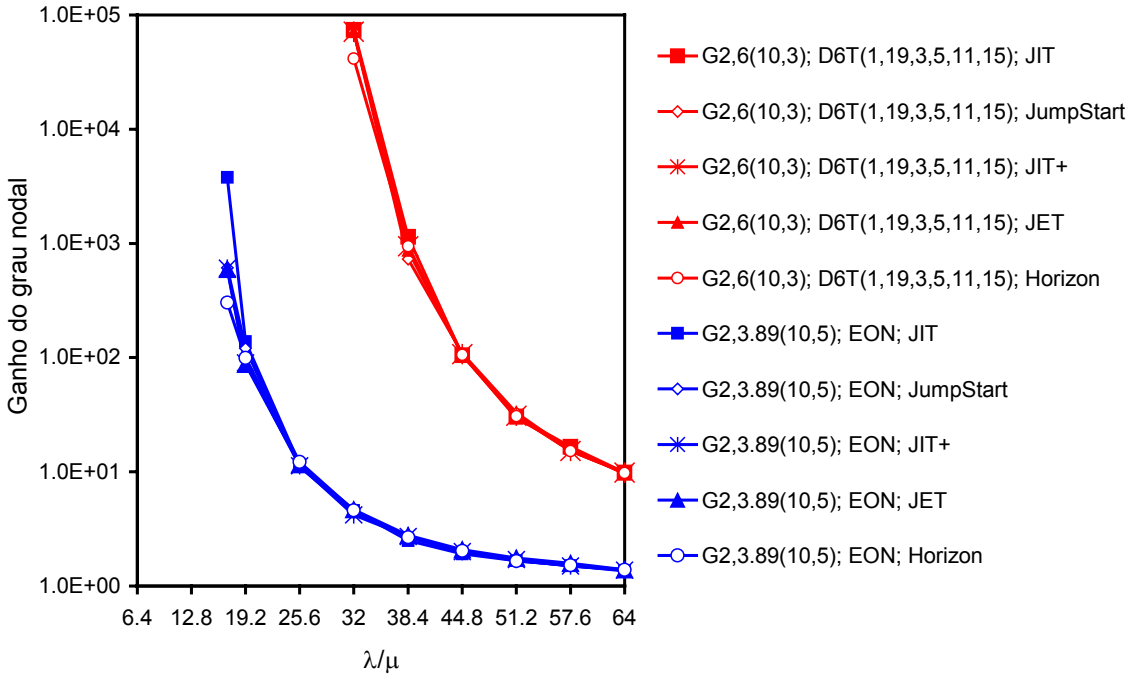


Fig. XX. Ganho do grau nodal dado o aumento do grau nodal de 2 (D2T(1,19)) para: 3.89 (EON) e 6 (D6T(1,19,3,5,11,15)), em função de λ/μ, no último salto de cada topologia, para os protocolos JIT, JumpStart, JIT$^+$, JET e Horizon; N=20; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

A Figura XXI exibe o ganho do grau nodal no último salto de cada topologia em função do grau nodal, incrementando o grau nodal de 2 (D2T(1,13)) para 3 (NSFNET (*N*=14)); de 2 (D2T(1,15)) para: 3 (D3T(1,15,5)), 3.125 (NSFNET (*N*=16)), 4 (D4T(1,15,5,13) e Mesh-Torus (*N*=16)), 5 (D5T(1,15,7,3,9)) e 6 (D6T(1,15,3,5,7,11)); de 2 (D2T(1,18)) para 3.89 (EON (*N*=19)); de 2 (D2T(1,19)) para: 3 (D3T(1,19,7)), 3.2 (ARPANET (*N*=20)), 4 (D4T(1,19,3,9)), 5 (D5T(1,19,3,7,11)), 6 (D6T(1,19,3,5,11,15)); de 2 (D2T(1,24)) para 4 (Mesh-Torus (*N*=25)) e de 2 (D2T(1,29)) para 6 (D6T(1,29,3,7,11,13)), para o protocolo JIT (*F*=64). Como se pode observar, quando o grau nodal aumenta de 2 para cerca de 3, o maior ganho é observado para o anel com cordas de grau três (ligeiramente menos que três ordens de grandeza) e o menor ganho verifica-se para a ARPANET (menor que uma ordem de grandeza). Quando o grau nodal aumenta de 2 para cerca de 4, o maior ganho é observado para o anel com cordas de grau quatro (com um ganho entre quatro e cinco ordens de grandeza) e o menor verifica-se para a rede óptica europeia (com um ganho menor que uma ordem de grandeza). Quando o grau nodal aumenta de 2 para cerca de 5 ou 6, o ganho cifra-se entre quatro e seis ordens de grandeza, dependendo do número de nós. Estes resultados confirmam claramente a importância do modo como são efectuadas as ligações numa rede OBS. As figuras com os restantes quatro protocolos não são mostradas neste resumo, pois apresentam resultados semelhantes.

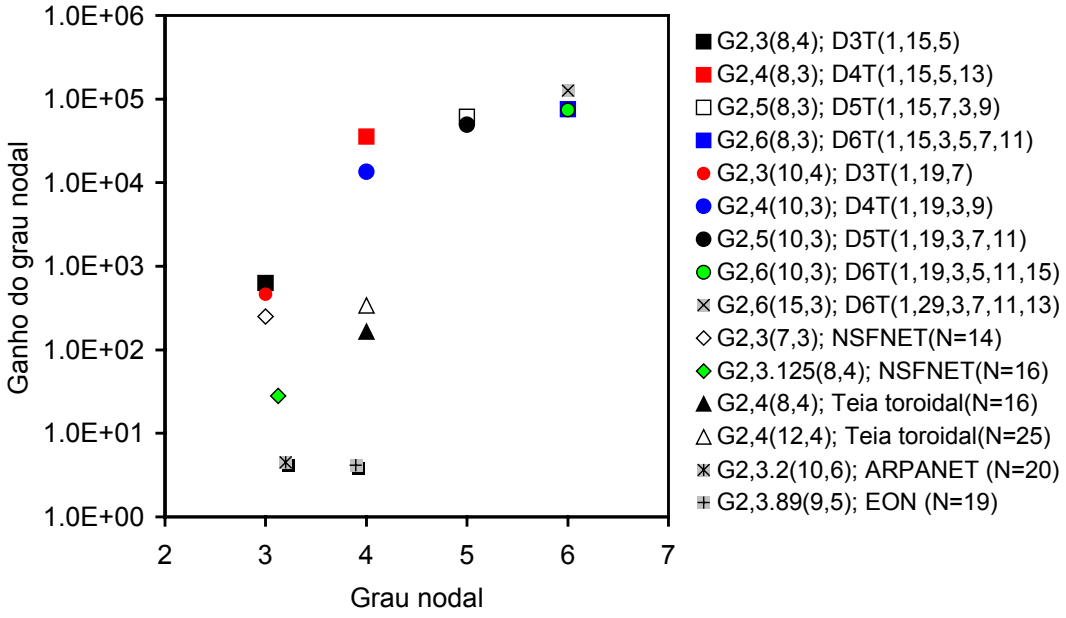


Fig. XXI. Ganho do grau nodal no último salto de cada topologia, em função do grau nodal para o protocolo de reserva de recursos JIT; λ/μ=32; *F*=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

Avaliação do Desempenho de Redes OBS usando o Protocolo E-JIT

O *Enhanced Just-in-Time* (E-JIT) é um protocolo de reserva de recursos proposto nesta tese e descrito anteriormente na secção relativa à descrição dos protocolos de reserva de recursos. Assim, neste momento, torna-se necessário avaliar o desempenho deste novo protocolo face aos existentes, já avaliados nas secções precedentes. Conforme foi verificado, o desempenho dos cinco protocolos em estudo (JIT, JumpStart, JIT$^+$, JET e Horizon) é bastante semelhante. Por outro lado, o E-JIT é um protocolo baseado no tradicional JIT. Assim, tendo em conta estas duas observações, esta secção estuda o desempenho do protocolo E-JIT em redes OBS em comparação com o JIT. O estudo é efectuado para redes em anel, anel com cordas de grau três e quatro, FCCN-NET, NSFNET (com 14 e 16 nós), teia toroidal (com 16 e 25 nós), ARPANET e rede óptica europeia.

A Figura XXII mostra a probabilidade de perda de *bursts* em função do número de canais de dados por ligação para os protocolos de reserva de recursos JIT e E-JIT, no último salto do anel (D2T(1,19)), anel com cordas (D3T(1,19,7) e D4T(1,19,3,9)), ARPANET (N=20) e rede óptica europeia (N=19) com λ/μ=32. As redes D3T(1,19,7) e D4T(1,19,3,9) são os anéis com cordas com o diâmetro mínimo de grau três e grau quatro, respectivamente, que apresentam melhor desempenho. Como se pode observar, esta figura confirma que os anéis com cordas com maior grau nodal têm um desempenho melhor relativamente a outras topologias. Também se observa que o desempenho do E-JIT é melhor que o do JIT, principalmente quando o valor da probabilidade de perda de *bursts* é menor. Outra observação prende-se com a semelhança do desempenho da ARPANET em comparação com a rede óptica europeia. A Figura XXIII confirma estes resultados, apresentando a probabilidade de perda de *bursts* em função do número de saltos das redes em anel (D2T(1,19)), anel com cordas ((D3T(1,19,7) e D4T(1,19,3,9)), ARPANET (N=20) e rede óptica europeia (N=19) para os protocolos JIT e E-JIT, com λ/μ=32. Apesar do diâmetro mínimo da rede óptica europeia ser menor que o da ARPANET, o seu desempenho é semelhante no último salto de cada uma. Estes resultados confirmam a importância do modo como as ligações entre nós devem ser efectuadas, uma vez que topologias com o mesmo número de nós têm desempenhos diferentes (vejam-se os anéis com cordas e a ARPANET). Nos anéis com cordas verifica-se que as topologias com menor diâmetro mínimo obtêm melhor desempenho. Em termos do desempenho dos dois protocolos em estudo, para o anel com cordas de grau três, o E-JIT tem melhor desempenho que o JIT em cerca de uma ordem de grandeza.

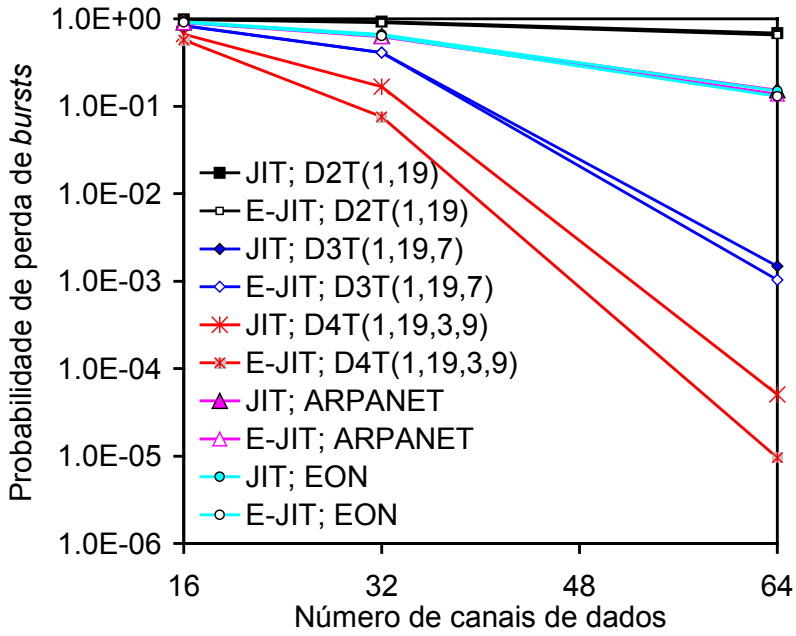


Fig. XXII. Probabilidade de perda de *bursts* em função do número de canais de dados por ligação (*F*), no último salto do anel (D2T(1,19)), anéis com cordas (D3T(1,19,7) e D4T(1,19,3,9)), ARPANET (*N*=20) e rede óptica europeia - EON (*N*=19) para os protocolos JIT e E-JIT; λ/μ=32; *N*=16; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

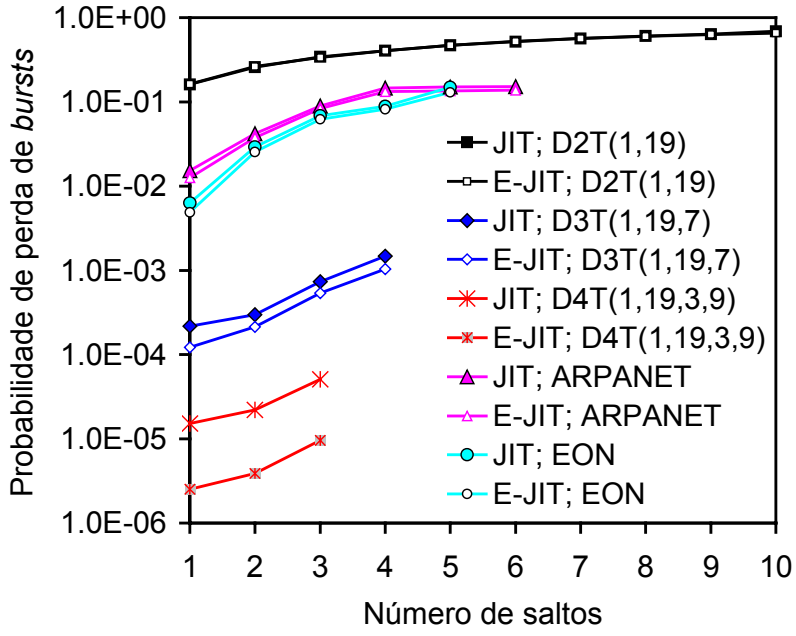


Fig. XXIII. Probabilidade de perda de *bursts* em função do número de saltos, para o anel (D2T(1,19)), anéis com cordas (D3T(1,19,7) e D4T(1,19,3,9)), ARPANET (*N*=20) e rede óptica europeia - EON (*N*=19) usando os protocolos JIT e E-JIT; λ/μ=32; *F*=64; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

As próximas figuras estudam o impacto do número de nós no desempenho dos protocolos de reserva de recursos JIT e E-JIT, considerando topologias em malha com o número de nós entre 10 e 30. A Figura XXIV inclui topologias de rede com grau nodal próximo de dois e três, e a Figura XXV contempla topologias de rede com grau nodal próximo de quatro. O anel com cordas de grau três (D3T(1,*N*-1,5)) é inserido em ambas as figuras para efeitos de comparação. A NSFNET com *N*=14 nós tem um grau nodal igual a 3, a NSFNET com *N*=16 nós tem um grau nodal igual a 3.125 e a ARPANET tem um grau nodal igual a 3.2 são exemplos de redes com grau nodal próximo de três. A teia toroidal com *N*=16 e *N*=25 e um grau nodal igual a 4 e a rede óptica europeia, com um grau nodal igual a 3.89 são exemplos de grau nodal próximo de quatro. Os anéis com cordas de diâmetro mínimo, que apresentam melhor desempenho, seleccionados para este estudo, são D3T(1,*N*-1,5) e D3T(1,*N*-1,7) para o grau três, e D4T(1,*N*-1,5,9) para o grau quatro.

Como se pode observar na Figura XXIV, para redes com o número de nós menor ou igual a 20 e igual a 22, D3T(1,*N*-1,5) tem melhor desempenho que outras topologias e D3T(1,*N*-1,7) tem melhor desempenho em redes com 20 nós e mais que 22. Para redes com o grau nodal próximo de três, o E-JIT apresenta um desempenho melhor que o JIT, mas o impacto não é significativo, excepto para D3T(1,15,5). Topologias de rede com grau nodal igual a dois (anéis e FCCN-NET) apresentam o pior desempenho. Em termos das outras redes irregulares, a NSFNET com 14 nós tem um desempenho melhor que a NSFNET com 16 nós e a ARPANET.

Na Figura XXV, as topologias de rede de grau quatro têm melhor desempenho em redes com mais de 16 nós. Em redes com grau nodal próximo de quatro, o E-JIT tem um desempenho melhor que o JIT, principalmente nos anéis com cordas de grau quatro com mais de 16 nós. No que concerne às topologias irregulares, o desempenho das teias toroidais (com *N*=16 e *N*=25 nós) é muito semelhante e melhor que o da rede óptica europeia.

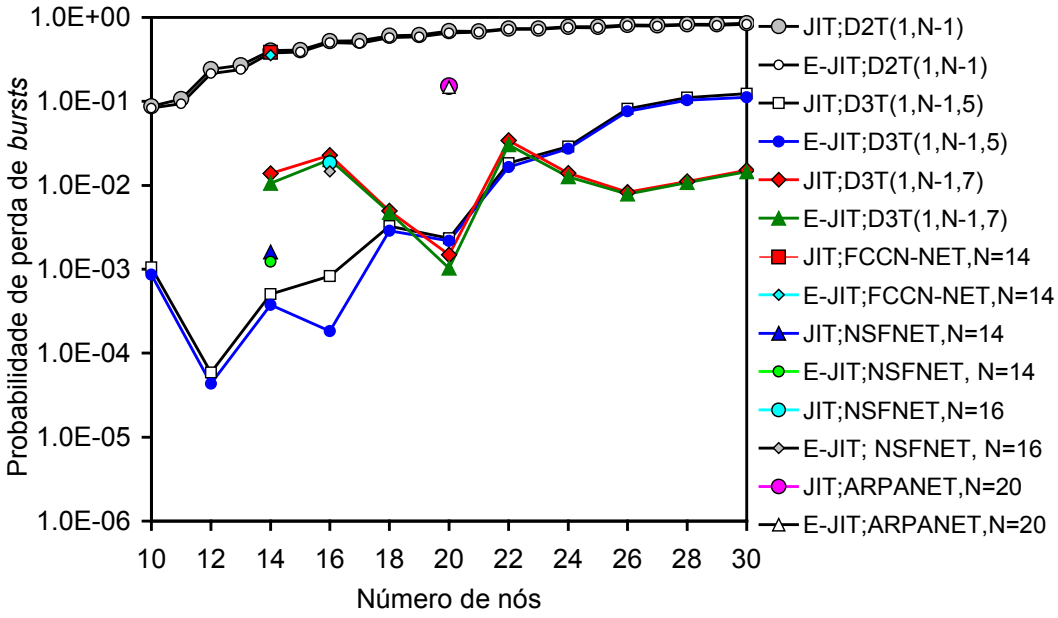


Fig. XXIV. Probabilidade de perda de *bursts* em função do número de nós (*N*), no último salto de anéis, anéis com cordas de grau três, FCCN-NET, NSFNET e ARPANET, para os protocolos JIT e E-JIT; λ/μ=32; *F*=64; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

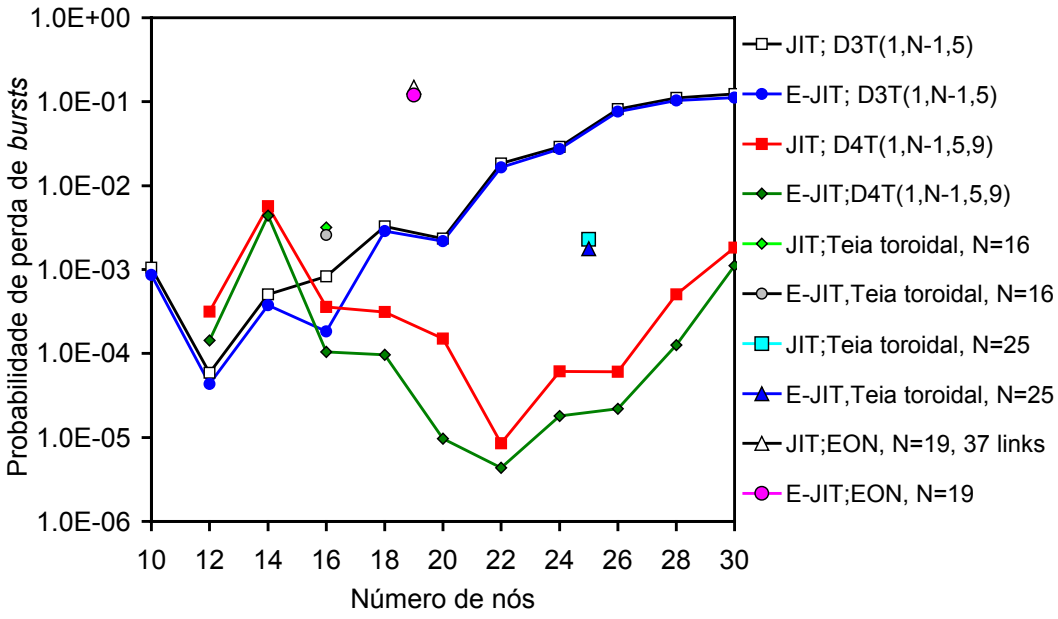


Fig. XXV. Probabilidade de perda de *bursts* em função do número de nós (*N*), no último salto de anéis, anéis com cordas de grau três e quatro, teias toroidais e rede óptica europeia, para os protocolos JIT e E-JIT; λ/μ=32; *F*=64; ; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

Antes de terminar, estuda-se também a influência do tempo de processamento da mensagem de *setup* (T_{Setup}) e tempo de configuração do OXC (T_{OXC}) no desempenho das redes OBS com topologias em malha para os protocolos JIT e E-JIT. A Figura XXVI apresenta a probabilidade de perda de *bursts* em função do tempo de configuração do OXC no último salto de D4T(1,19,3,9), teia toroidal (N=16), teia toroidal (N=25) e rede óptica europeia para os protocolos JIT e E-JIT, com F=64 e λ/μ=44.8. Nesta figura, é definido um valor fixo de T_{Setup} para o JIT e o E-JIT (T_{Setup}(JIT)=T_{Setup}(E-JIT)=12.5μs), tendo em conta a tecnologia actualmente disponível (os controladores JITPAC [141]). Assume-se que o T_{OXC} varia entre um valor estimado para um cenário de futuro próximo (T_{OXC} =20μs) e até dez vezes o valor definido para a tecnologia actualmente disponível, isto é, T_{OXC}=10*10ms=100ms. Como se pode observar, os anéis com cordas de grau quatro apresentam um desempenho melhor que as outras topologias. O pior desempenho verifica-se para a rede óptica europeia. As teias toroidais têm um desempenho semelhante para os dois números de nós considerados. Este resultado pode ser explicado pelo mesmo grau nodal (quatro) e pelo mesmo modo como os nós são interligados entre si. Para valores de T_{OXC} menores que 1ms, não se verifica melhoria do desempenho de cada rede. Esta observação confirma resultados anteriores, onde se constata que o desempenho das redes é independente do valor de T_{OXC}, quando este apresenta valores inferiores a 1ms. Adicionalmente, pode-se observar que o desempenho relativo dos dois protocolos de reserva de recursos é similar, sendo, no entanto, o desempenho do E-JIT sempre melhor que o do JIT. Este resultado torna-se mais evidente, sobretudo para D4T(1,19,3,9), quando $T_{OXC}\leq$10ms. Estes resultados são confirmados pelas Figuras XXVII e XXVIII.

A Figura XXVII mostra a probabilidade de perda de *bursts* em função do tempo de configuração do OXC no último salto de D4T(1,19,3,9), teia toroidal (N=16), teia toroidal (N=25) e rede óptica europeia para os protocolos JIT e E-JIT, com o T_{Setup} a variar de acordo com (EA.4), F=64 e λ/μ=44.8. Por seu turno, a Figura XXVIII exibe a probabilidade de perda de *bursts* em função do tempo de processamento da mensagem de *setup* no último salto das topologias de rede OBS consideradas nas duas figuras anteriores, para os protocolos de reserva de recursos JIT e E-JIT, com F=64 e λ/μ=44.8. O T_{Setup} assume valores entre 1μs e 12.5μs, considerando 5μs e 10μs como tempos intermédios. Quanto ao T_{OXC}, este assume valores que variam em função do

T_{Setup} calculados de acordo com (EA.7). Como se pode observar, tanto esta figura como a anterior confirmam as observações efectuadas para a Figura XXVI.

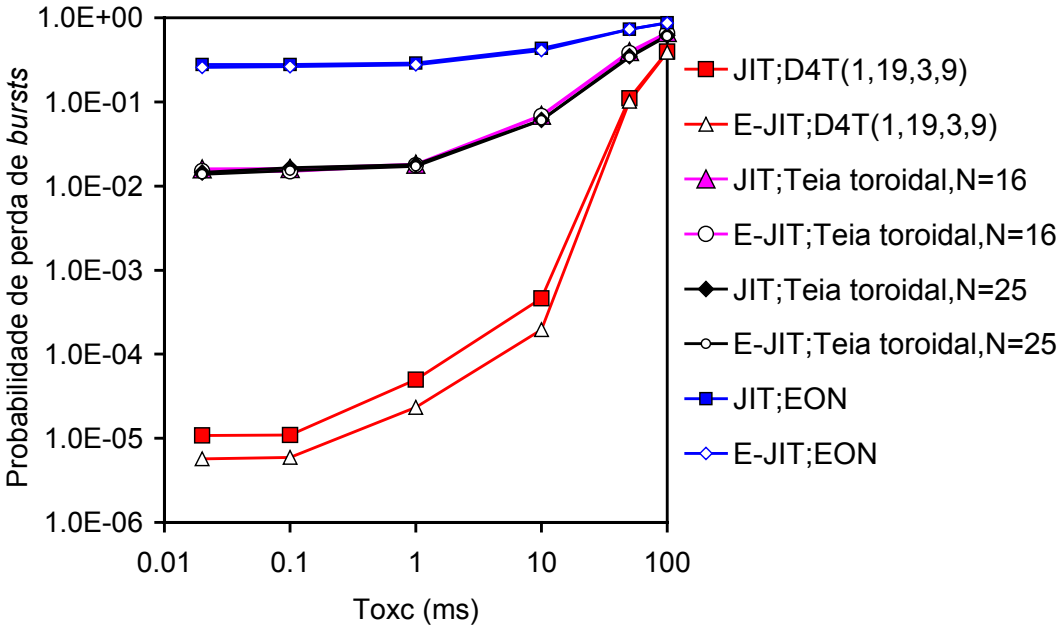


Fig. XXVI. Probabilidade de perda de bursts em função do tempo de configuração do OXC no último salto de D4T(1,19,3,9), teia toroidal (N=16), teia toroidal (N=25) e rede óptica europeia para o JIT e o E-JIT; F=64; λ/μ=44.8; T_{Setup}(JIT)=T_{Setup}(E-JIT)=12.5μs.

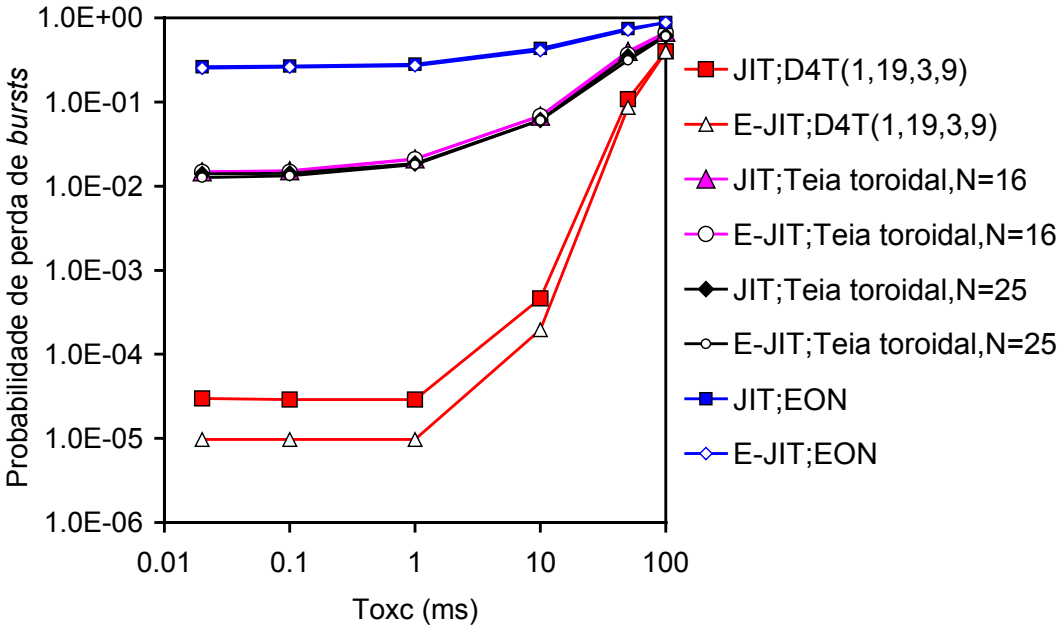


Fig. XXVII. Probabilidade de perda de *bursts* em função do tempo de configuração do OXC no último salto de D4T(1,19,3,9), teia toroidal (N=16), teia toroidal (N=25) e rede óptica europeia para o JIT e o E-JIT; F=64; λ/μ=32; com o T_{Setup} a variar de acordo com (EA.4) para cada protocolo.

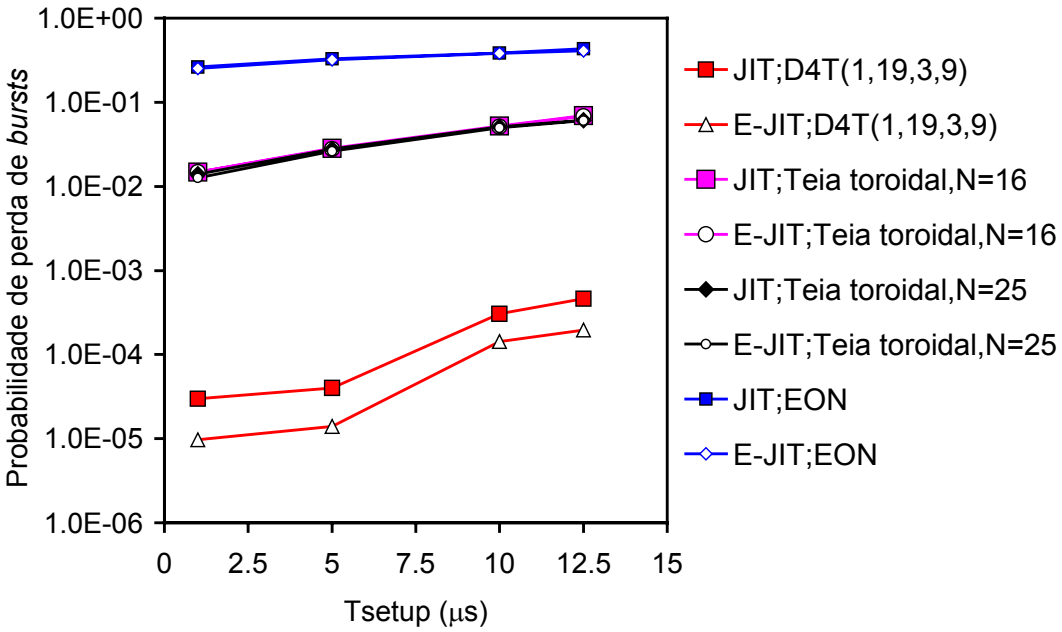


Fig. XXVIII. Probabilidade de perda de *bursts versus* T_{Setup} no último salto de D4T(1,19,3,9), teia toroidal (N=16), teia toroidal (N=25) e rede óptica europeia para o JIT e o E-JIT; F=64; λ/μ=32, com o T_{OXC} a variar de acordo com (EA.7), para cada protocolo.

Conclusões e Sugestões para Trabalho Futuro

Ao longo da presente tese foi estudado o desempenho de protocolos de reserva unidireccional de recursos em redes IP sobre infra-estruturas com comutação óptica de agregados de pacotes (redes OBS). Este ponto apresenta uma síntese da tese e aponta algumas direcções para trabalho futuro.

Delimitando o tema da tese, e apresentadas as suas contribuições, foram descritos os principais aspectos das redes baseadas no paradigma de comutação óptica de agregados de pacotes. Neste ponto, começou-se por descrever a arquitectura das redes OBS, prestando especial atenção aos nós fronteira e aos nós de interligação. Depois, o trabalho centrou-se nas principais tarefas desempenhadas pelos nós fronteira (principalmente, o processo de agregação em *bursts* e os respectivos algoritmos mais importantes). Em relação aos nós de interligação, foram descritos os protocolos de reserva de recursos, onde se incluiu a proposta de um novo protocolo de reserva unidireccional de recursos, designado por *Enhanced Just-in-Time* (E-JIT), e o problema da resolução da contenção, abordando os principais esquemas para resolver este problema.

Em seguida, foram apresentados os objectivos, o desenho, implementação e validação de um simulador para redes OBS, designado por OBSim. Após analisar as metodologias existentes para avaliação do desempenho de redes OBS, foi identificada a necessidade de desenvolver um simulador de raiz. Este simulador implementou um modelo baseado em objectos que permite estimar o desempenho de uma rede OBS definida pelo utilizador. Os resultados do simulador foram validados e, doravante, este simulador pode ser usado como uma ferramenta para estimar o desempenho de redes OBS.

Foi referida cada uma das topologias em malha estudadas e utilizadas para avaliar o desempenho de redes OBS, tanto regulares como irregulares. As topologias regulares consideradas foram anéis, anéis com cordas (de grau três a seis) e teia toroidal (com 16 nós e 32 ligações, e com 25 nós e 50 ligações). No que concerne a topologias irregulares foram seleccionadas as seguintes: NSFNET (com 14 e 16 nós), ARPANET, rede óptica europeia, e a proposta apresentada para a rede de interligação da Fundação para a Computação Científica Nacional (FCCN-NET).

A seguir, foi efectuada a avaliação de desempenho de redes OBS com protocolos de reserva unidireccional de recursos. Foram considerados os cinco protocolos mais relevantes propostos na literatura (JIT, JumpStart, JIT$^+$, JET e Horizon) e o protocolo proposto E-JIT, para redes OBS com topologias em malha.

Para avaliar o desempenho das redes OBS foram definidas a probabilidade de perda de *bursts*, o ganho do grau nodal e o ganho do comprimento da corda. A probabilidade de perda de *bursts* é uma métrica muito importante para avaliar o desempenho das redes OBS e foi definida como a probabilidade da transmissão de um *burst* não atingir o seu destino. O ganho do grau nodal e o ganho do comprimento da corda foram propostos com o objectivo, respectivamente, de quantificar os benefícios em termos do incremento do grau nodal e de quantificar os benefícios que advêm da escolha do melhor comprimento da corda.

O estudo centrou-se na avaliação do desempenho de anéis e anéis com cordas de grau três e quatro, para redes com 20 nós. Verificou-se que os anéis com cordas de diâmetro mínimo de grau quatro (D4T(1,19,3,9)) têm melhor desempenho que as topologias com menor grau nodal e que outras topologias com o mesmo grau nodal e diferente comprimento da corda (w_4). Quanto ao ganho do grau nodal, aumentando o grau nodal de dois (anel) para três (anel com cordas de grau três), é entre duas e três ordens de grandeza no último salto, ao passo que, incrementando o grau nodal de dois para quatro (anel com cordas de grau quatro), o ganho obtido é à volta de quatro ordens de grandeza. No estudo relativo ao ganho do comprimento da corda,

para topologias de grau três (D3T(1,19,w_3)), verificou-se que o maior ganho do comprimento da corda é ligeiramente menor que duas ordens de grandeza e as topologias com o comprimento de corda de w_3=5 e w_3=7 conduzem a melhores desempenhos, sendo o desempenho de w_3^*=7 ligeiramente melhor que o desempenho para w_3^*=5. O efeito do tempo de processamento da mensagem de setup (T_{Setup}) e do tempo de configuração do *cross-connect* óptico (T_{OXC}) também foi estudado. Os resultados mostraram que, para valores de $T_{OXC} \leq$1ms, o desempenho dos anéis com cordas é independente da variação do valor de T_{OXC}. Isto permite concluir que, mesmo com a redução do T_{OXC} para valores inferiores a 1ms não melhoram o desempenho da rede. Também se pode observar que para os anéis com cordas de grau quatro, uma redução do T_{OXC} de 10ms até 20μs conduz a uma melhoria de desempenho a rondar três ordens de grandeza. Para anéis com cordas de grau três, a redução do T_{OXC} leva a uma melhoria de desempenho à volta de duas ordens de grandeza. A redução do valor do T_{Setup} não apresentou uma melhoria significativa no desempenho das redes. Verificou-se, ainda, que a influência do grau nodal no desempenho de uma dada topologia é maior que a influência do T_{Setup} e do T_{OXC}. Para todos os casos estudados, o desempenho dos protocolos JIT, JumpStart, JIT$^+$, JET e Horizon é muito semelhante.

Depois, avaliou-se o desempenho dos cinco protocolos considerados, nas topologias em malha referidas anteriormente, incluindo anéis e anéis com cordas com o número de nós a variar entre 10 e 30. Quando o número de nós é menor que 16, as topologias D3T(1,*N*-1,5) têm melhor desempenho, ao passo que, quando o número de nós é superior a 16, D4T(1,*N*-1,5,9) acrescentam melhor desempenho. Verificou-se que topologias com grau nodal semelhante (como a NSFNET com 14 nós e o correspondente anel com cordas de grau três) apresentam desempenhos diferentes, sendo os melhores resultados obtidos para os anéis com cordas. Isto revela a importância do modo como as ligações são efectuadas entre os nós. O estudo confirma os resultados anteriores no que concerne à similaridade do desempenho dos cinco protocolos em estudo.

Em seguida, foi estudado o impacto do grau nodal no ganho do grau nodal em redes com 16 e 20 nós. Para redes com 16 nós, verificou-se que quando o grau nodal aumenta de 2 para 3, o maior ganho a rondar três ordens de grandeza é obtido para D3T(1,15,5) e o menor é apresentado pela NSFNET. Quando o grau nodal aumenta de 2 para 4, o maior ganho, entre quatro e cinco ordens de grandeza, verifica-se para

D4T(1,15,5,13) e o menor, entre duas e três ordens de grandeza, é dado pela teia toroidal. Quando o grau nodal aumenta de 2 para 5 (D5T(1,15,7,3,9)) ou 6 (D6T(1,15,3,5,7,11)), o ganho é à volta de cinco ordens de grandeza no último salto de cada topologia. Deste modo, o melhor desempenho, em termos do ganho do grau nodal, para todas as topologias com 16 nós estudadas foi o anel com cordas de grau seis. Para redes com 20 nós foi possível observar que, quando o grau nodal aumentou de 2 para à volta de 3, o maior ganho é obtido pelo anel com cordas de grau três (D3T(1,19,7)), sendo ligeiramente menor que três ordens de grandeza, e o menor ganho ocorre para a EON, sendo o seu ganho menor que uma ordem de grandeza. Quando o grau nodal aumenta de 2 para 4 (D4T(1,19,3,9)), o ganho varia entre uma e duas ordens de grandeza. Quando o grau nodal aumenta de 2 para 5 (D5T(1,19,3,7,11)) ou 6 (D6T(1,19,3,5,11,15)), o ganho é entre quatro e cinco ordens de grandeza.

Para estudar a influência do grau nodal no ganho do grau nodal, foram consideradas topologias em malha com grau nodal entre três e seis com número de nós compreendido entre 14 e 30. Quando o grau nodal aumenta de 2 para à volta de 3, o maior ganho observa-se para o anel com cordas de grau três (ligeiramente menos que três ordens de grandeza) e o menor ganho verifica-se para a ARPANET (com menos que uma ordem de grandeza). Quando o grau nodal aumenta de 2 para cerca de 4, o maior ganho constata-se para o anel com cordas de grau quatro (com um ganho compreendido entre quatro e cinco ordens de grandeza) e o menor ganho verifica-se para a EON (com menos que uma ordem de grandeza). Quando o grau nodal aumenta de 2 para 5 ou 6, o ganho situa-se entre quatro e seis ordens de grandeza, dependendo do número de nós. Estes resultados confirmam observações anteriores no que concerne à importância do modo como os nós estão interligados entre si. Também se confirma a proximidade de desempenho em relação aos cinco protocolos estudados. Deste modo, pode-se concluir que a utilização dos protocolos com reserva imediata (JIT, JumpStart e JIT^{+}) são mais apelativos para utilização, uma vez que a sua implementação é mais simples que a dos protocolos com reserva atrasada.

Finalmente, foi efectuada a avaliação do desempenho do protocolo E-JIT em comparação com o desempenho do JIT. Este estudo foi realizado em comparação com o JIT com o objectivo de aumentar a legibilidade das figuras, reduzindo o número de curvas, uma vez que o desempenho relativo dos cinco protocolos propostos na literatura é muito semelhante, conforme foi referido anteriormente. Além disso, o E-JIT é um protocolo baseado no JIT e mantém toda a sua simplicidade em termos de implementação. Deste modo, para redes com 20 nós, observou-se que,

quando a probabilidade de perda de *bursts* é menor, o desempenho relativo do E-JIT é mais significativo. Em termos do ganho do grau nodal, da influência do tempo de processamento da mensagem de setup e do tempo de configuração do OXC, verifica-se que os resultados são semelhantes aos obtidos e analisados anteriormente para os outros protocolos, sendo o desempenho do E-JIT melhor que o do JIT, principalmente quando a probabilidade de perda de *bursts* é menor.

O principal objectivo desta tese visava a apresentação de um estudo do desempenho de redes OBS com topologias em malha e em anel para os mais importantes protocolos de reserva unidireccional de recursos (JIT, JumpStart, JIT$^+$, JET e Horizon). Isto foi conseguido tendo em conta o tempo de atraso entre a mensagem de *setup* e o respectivo *burst*, o tempo de atraso entre os nós fronteira e o respectivo nó de interligação, o tempo de propagação ao longo do meio óptico de comunicação entre nós de interligação, o número de canais de dados por ligação, o tempo de processamento da mensagem de *setup*, o tempo de configuração do comutador óptico, a dimensão e a topologia da rede. Este objectivo foi atingido com sucesso. Além disso, como resultado desta investigação, foi possível desenvolver uma optimização na operação do JIT, o que levou à proposta de um novo protocolo de reserva unidireccional de recursos chamado *Enhanced Just-in-Time* (E-JIT). Estes resultados consideram-se relevantes no âmbito do desenvolvimento da próxima geração da Internet óptica.

Sugestões para Trabalho Futuro

Ao terminar esta tese, resta sugerir futuros temas de investigação resultantes do trabalho de investigação desenvolvido:

- Estudar aspectos de qualidade de serviço relacionados com os protocolos de reserva unidireccional de recursos analisados nesta tese (JIT, JumpStart, JIT$^+$, JET Horizon e E-JIT). Por exemplo, introduzir diferentes prioridades para diferentes classes de tráfego.
- Investigar o efeito dos conversores de comprimento de onda nos protocolos de reserva unidireccional de recursos examinados nesta tese.
- Estudar diferentes configurações da FCCN-NET, nomeadamente, reduzindo o número de nós e, principalmente, o número de nós fronteira por nó de interligação.
- Implementar e avaliar o desempenho do E-JIT numa bancada de testes ou numa rede OBS real.

Chapter 1

Introduction

1.1 Focus

A fiber optic network consists of nodes connected by optical fibers to transport data over those optical fibers. Fiber optic networks may be classified according to their evolution as first-generation and second-generation networks [34, 35]. First-generation fiber optic networks focused on solving the link bottleneck problem and, therefore, they use optical fiber as a transmission medium, but every switching, processing and routing functions are done at the electronic level. At present, these networks are widely used in every kind of telecommunication networks, with the possible exception of access networks and small local area networks. Examples of first-generation fiber optic networks include *i*) SONET (Synchronous Optical Network) and SDH (Synchronous Digital Hierarchy) networks, which form the core of the telecommunications infrastructure in North America, Europe and Asia; *ii*) the fiber optic variety of Ethernet (10 Mbps), Fast Ethernet (100 Mbps), and Gigabit Ethernet (1 Gbps) networks, widely used in local area networks (LANs); *iii*) the FDDI (Fiber Distributed Data Interface) rings which are being replaced by Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet in LANs and metropolitan area networks (MANs); *iv*) DQDB (Distributed Queue Dual Bus) used in MANs which is

being replaced by Gigabit Ethernet and 10-Gigabit Ethernet; and *v*) a variety of storage area networks (SANs) such as ESCON (Enterprise Serial Connection), HIPPI (High Performance Parallel Interface), Fibre Channel, FICON (for *Fiber Connectivity*), and InfiniBand.

Recently, it has been recognized that fiber optic networks are capable of providing more functions than just point-to-point transmission. Therefore, second-generation fiber optic networks are being focused on solving the router bottleneck problem [34]. In first-generation networks, a given electronic node must handle not only the data destined for that node, but also the data that is destined to other nodes. If data destined for other network nodes could be routed through the optical domain, the load on the electronic part at that node would be significantly reduced. This is one of the key driving issues for the development of second-generation fiber optic networks. This kind of networks, using wavelength division multiplexing (WDM) and wavelength routing, is now being deployed.

A third-generation of fiber optic networks, also known as next generation optical Internet networks, is also emerging [36-38]. In this new generation, the interoperability between the WDM layer and the IP (Internet protocol) layer[6] is the major concern, which leads to both lower management costs and lower complexity. It consists in the use of two-layer architecture, in which IP traffic is transported directly over optical networks, as shown in Fig. 1.1.

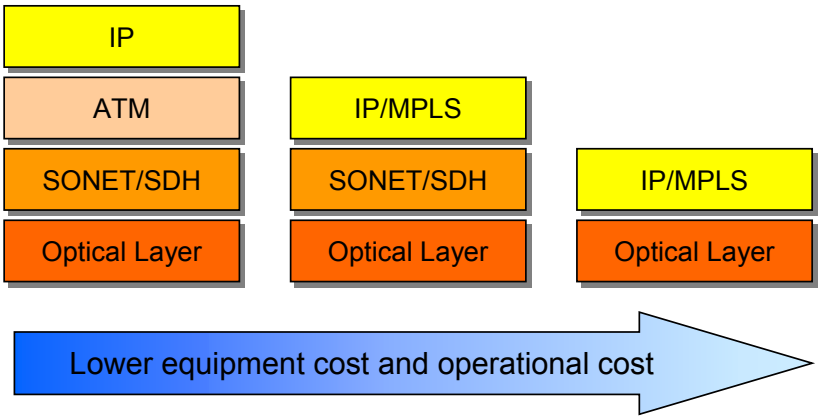


Fig. 1.1. Evolution toward a two-layer architecture.

[6] There is an overuse of these terms in the literature: in fact, IP is the main protocol of the TCP/IP network layer.

As referred above, nowadays, data are transmitted optically in wide area networks (WANs), MANs, and in some LANs. In terms of switching technologies for optical networks there are two main types: electronic switching and optical switching. In short, electronic switching uses digital (electronic) switching fabrics and converts data from optical to electronic, for switching, and then from electronic to optical for transmission. On the other hand, optical (photonic) switching uses photonic switching fabrics and keeps data in the optical domain. In this thesis, only optical switching is considered.

In order to make the use of optical fibers more flexible and to allow a better use of the huge potential capacity of optical fibers, some paradigms have been proposed for optical switching, namely optical circuit switching [39-43], optical packet switching [37, 43-53], and optical burst switching [5, 37, 43, 47, 52, 54-60].

In optical circuit switching [39-43] (via wavelength routing in WDM networks), a lightpath (also referred to as λ-channel) between the source node and the destination node, using a dedicated wavelength on each link along a physical path, is established. If converters are present, a lightpath may consist of different wavelengths along the path. In this switching paradigm, there are three distinct phases: circuit set-up, data transfer, and circuit tear-down [43]. In the first phase, only control messages are exchanged to set-up an end-to-end circuit between source and destination nodes (ingress and egress nodes, respectively). This virtual circuit uses a dedicated channel of a fixed bandwidth (e.g. a time slot or a frequency), from the source to the destination. Data transfer characterizes the second phase. When data transfer is complete, the circuit is released – third phase. For Internet services using optical circuit switching, the bandwidth would not be efficiently used because this kind of applications does not have a long duration data transmission relative to the set-up time of the lightpath [57].

In optical packet switching [37, 43-53, 61] data can be sent without setting up a circuit and data are transmitted in optical packets. Each packet contains a header with some control information (in-band control) and it is sent to the destination via intermediate nodes in a 'store-and-forward' technique. In this technique, when a packet arrives at a node, it is stored first and then, after the processing of the header, is forwarded to the next node. This implies that the switch is configured only after the arrival of the packet. Packets can have either fixed (e.g. Asynchronous Transfer Mode - ATM - cell) or variable size (e.g. IP packet) with a limited maximum size. In this switching paradigm, no optical-to-electrical or electrical-to-optical conversions are required.

However, presently, technology for optical packet switching is immature due mainly to the three following problems [46, 47, 52, 62]:

- Synchronization — optical packet switches usually work synchronously. For example, if packets arrive at different input ports, it is necessary to align them before entering switch fabric. However, it is both difficult and expensive to develop an optical synchronization module [46].
- Optical buffering — in optical packet switching it is necessary store-and-forward packets. This technique is required due to output port contention. However, at present, optical random access memory (RAM) is still immature, and optical buffering is accomplished by using fiber delay lines (FDLs). FDLs give a fixed delay that depends on the fiber length [47, 52].
- Time to configure optical switch fabric — given the current state of the art, for an optical packet switch, the optimistic value of 1*ms* is needed to set-up a connection from an input port to an output port. For example, following [62], at a data bit rate of 2.5 Gbps, it takes about 5μs to transmit a 1500-byte packet. Thus, using a switch that operates at a packet level, less than 0.5% of the time is used for switching data while the rest is spent in setting up the switch fabric.

Optical burst switching (OBS) [5, 37, 43, 47, 52, 54-60, 63-65] is a technical compromise between static wavelength routing (i.e. optical circuit switching) and optical packet switching, since it does not require optical buffering or packet-level parsing, and it is more efficient than circuit switching if the traffic volume does not require a full wavelength channel. In OBS networks, IP packets are assembled into very large size packets called data bursts. These bursts are transmitted after a burst header packet (control packet), with a delay of some offset time. Control and data packets are transmitted separately in different channels (also called wavelengths or Lambdas – λ's)[7]. Each burst header packet contains routing and scheduling information and it is processed at the electronic level, before the arrival of the corresponding data burst. OBS has not yet been standardized, but it is considered as

[7] As far as OBS are concerned, the words channel and wavelength (or λ) are used interchangeable.

a viable solution to the problem of bursty traffic over an optical network [66]. Therefore, it will be discussed in detail in Chapter 2, based on published research literature.

Table 1.1 summarizes a comparison of all-optical switching paradigms [52, 57, 64, 164, 165].

Optical Switching Paradigms	**Optical Circuit Switching**	**Optical Packet Switching**	**Optical Burst Switching**
Control	Out-of-band	In-band	Out-of-band
Switching Unit	λ-channel /fiber	Packet	Burst
Bandwidth Utilization	Low	High	High
Wavelength Conversion	Not necessary	Necessary	Necessary
Latency (set-up)	High	Low	Low
Optical Buffer	Not required	Required	Not required
Switching Speed Requirement	Slow	Fast	Medium
Traffic Adaptivity (e.g. fault)	Low	High	High

Table 1.1. A comparison of different optical switching paradigms.

Due to the exponential-like increase of the Internet traffic, optical packet switching (OPS) is also gaining more interest [67-69]. However, OPS is still an immature technology, mainly due to the lack of optical random access memory (RAM) and due to the problems with synchronization as mentioned above [47, 52]. Nevertheless, several approaches have been proposed to solve this problem, namely the framework conducted by Internet Engineering Task Force (IETF) [70], i.e. the Generalized MultiProtocol Label Switching (GMPLS) [71, 72], and the Optical Burst Switching (OBS) [1, 57, 58, 63, 64, 73-75].

Generalized MultiProtocol Label Switching (GMPLS) extends the label switching architecture proposed in MultiProtocol Label Switching (MPLS) in order to

include other types of non-packet networks, such as wavelength routing networks and SONET/SDH. GMPLS defines four types of interfaces [37]: *i*) packet switch capable; *ii*) time division multiplex; *iii*) lambda switch capable; and *iv*) fiber switch capable. These interfaces forward data based on the content, respectively [37]: *i*) packet\/cell header; *ii*) time slot of data; *iii*) incoming wavelength; and *iv*) incoming fiber. GMPLS extends the control plane of MPLS to support each of these four interfaces. The extension of MPLS to optical networks helps to minimize the cost of transition from the wavelength routing technology to either the optical burst switching or optical packet switching technologies.

GMPLS allows the direct integration of the IP layer with the optical layer. Moreover, it can be applied to wavelength routing networks, optical burst switched networks, and optical packet switched networks. A WDM network using MPLS can be classified as [76]: labeled optical circuit switching, labeled optical burst switching, and labeled optical packet switching, depending on the switching technology used. The labeled optical circuit switching is also referred to as lambda labeling or multiprotocol lambda switching (MPλS) [77-79].

This thesis focuses on optical burst switching networks. As described above, in this kind of network, the transmission of a data burst (in a given data channel) occurs after the transmission of a setup message (in the control channel). This setup message contains routing and scheduling information to be processed by each intermediate switch in order to configure the optical switch fabric to switch the corresponding data burst to the suitable output port. Several one-way resource reservation protocols have been proposed for optical burst switched networks. Two types of reservation schemes can be considered: immediate and delayed. Just-in-time (JIT) [74, 75] and JIT$^+$ [80] resource reservation protocols are examples of immediate reservation, whereas just-enough-time (JET) [57, 73, 81] and Horizon [1, 54] are examples of delayed reservation protocols. JumpStart [75, 82-86] is an example of OBS signaling protocol, proposed from the JIT protocol, which can be configured as an immediate or delayed reservation scheme. Along this thesis, particular attention will be paid to the performance of these resource reservation protocols in OBS networks with ring and mesh topologies.

1.2 Problem Definition and Objectives

At the starting point of this research program, in September of 2002, considerable attention had been paid to OBS networks. However, most of the published studies were theoretical in nature [1-6] and typically considered very simple OBS networks being usually analyzed an output port of an OBS node. Therefore, several network parameters such as the network size, the network topology, the number of channels per link, etc, were usually ignored and new methodologies were needed for performance assessment of this kind of network. Moreover, most of the studies, reported up to that time, ignored several important OBS parameters such as the burst offset length, the processing time of setup messages and the optical switch configuration time, which may have a significant impact on the network performance. Under these circumstances, previous studies have shown that the JIT protocol had a worse performance, in terms of burst loss probability, than either JET or Horizon. In fact, the advanced scheduling and void filling algorithms used in JET and Horizon might lead, at a first thought, to the idea that JET and Horizon should have better performance than JIT. However, it is reasonable to assume that the processing of setup messages for JET and Horizon will be longer than for JIT due to complex operations and/or large number of memory lookups, and therefore, it is not clear whether the more efficient scheduling algorithms will compensate the higher processing overhead incurred. On the other hand, if the optical switch configuration time is much longer than the mean burst length, the impact of the scheduling efficiency on the burst loss probability may be small. Moreover, cascading effects due to the fact of having several optical switches in the optical path need to be investigated since each switch needs to be configured by the setup message before handling the burst. Besides, there has been an increasing interest in mesh topologies for optical WANs [7-16]. Therefore, there is a need for more detailed performance studies taking into account the burst offset length, the processing time of setup messages, the optical switch configuration time and optical switch cascading effects in OBS networks with ring or mesh topologies.

The main objective of this thesis is to present a performance study of OBS networks with ring and mesh topologies for the most important five one-way resource reservation protocols (JIT, JumpStart, JIT$^+$, JET, and Horizon), taking into account the burst offset length, the edge to core node delay, the propagation delay on each

link between core nodes, the number of data channels per link, the processing time of setup messages, the optical switch configuration time, the network size, and the topology.

To reach this objective the following intermediate objectives were identified and performed:

- Detailed analysis of the OBS network architecture;
- Detailed analysis of the most important five one-way resource reservation protocols;
- Proposal, implementation and validation of an object-oriented model for simulation of IP over OBS networks with any regular or irregular topology and for the five one-way resource reservation protocols taking into account the previously ignored OBS parameters such as burst offset length, the processing time of setup messages and the optical switch configuration time;
- Performance study of the five one-way resource reservation protocols in IP over OBS networks with ring and chordal ring topologies;
- Performance study of the five one-way resource reservation protocols in IP over OBS networks with mesh topologies.

Although it was not an initial objective, after the analysis of the above-mentioned five one-way resource reservation protocols, a possible optimization in the operation of JIT was identified, which led to a proposal of a new one-way resource reservation protocol called Enhanced JIT (E-JIT). The performance of this new protocol was also evaluated and compared with existing JIT, using the simulation tool developed in the scope of this work. With this study it was concluded that E-JIT performs better than JIT confirming the importance of the setup message processing time on the performance of these protocols.

1.3 Thesis organization

This thesis is organized in six chapters and the chapters are organised as follows. This chapter, the first, presents the context of the thesis, focusing the topic under studies, the definition of the problem and main objectives, the thesis organization and its main contributions.

Chapter 2 presents the state of the art of the optical burst switched (OBS) networks. This chapter begin with the description of the OBS network architecture being paid special attention to the edge and core nodes. After, it focuses on the main task performed by edge nodes, i.e., the burst assembly process. Concerning core nodes, Section 2.4 explain OBS resource reservation protocols, including the proposal of a new one-way resource reservation protocol, called enhanced just-in-time (E-JIT), and Section 2.5 focuses on the problem of contention resolution. Other OBS issues are also considered, namely, quality of service, transmission control protocol (TCP) over OBS, multicasting, burst grooming, and OBS applications.

In chapter 3, the design, implementation and validation of a simulator for OBS networks, named OBSim is described. The network topologies used to simulate the performance of OBS networks is also plotted.

Chapter 4 presents a performance assessment of JIT, JumpStart, JIT$^+$, JET, and Horizon one-way resource reservation protocols in OBS networks with ring and chordal ring topologies. This chapter proposes and describes the performance metrics used to evaluate OBS networks, i.e., the burst loss probability, nodal degree gain and chord length gain. In terms of chordal rings, it focuses on degree-three and degree-four chordal rings given the proximity with nodal degree of real network topologies like NSFNET, ARPANET or European Optical Network.

In Chapter 5, first, a performance assessment of JIT, JumpStart, JIT$^+$, JET, and Horizon resource reservation protocols in OBS networks with mesh topologies is shown. Later, a performance evaluation of E-JIT protocol is presented. The analysis is focused on OBS networks with the following mesh topologies: chordal rings with number of nodes ranging from 10 up to 30, mesh-torus with 16 and 25 nodes, the NSFNET with 14 nodes and 21 links, the NSFNET with 16 nodes and 25 links, the ARPANET with 20 nodes and 32 links, the European Optical Network (EON) with 19 nodes and 37 links, and the Portuguese *Fundação para a Computação Científica Nacional* network (FCCN-NET) with 14 nodes and 14 links.

Finally, Chapter 6 summarizes the main research findings and puts forward proposals for further research.

1.4 Main contributions

This section is devoted to the main scientific contributions of this thesis. Thus, the following paragraphs describe, in the opinion of the Author, main contributions for the advance of the state-of-art in the area of the optical Internet.

The first contribution of this thesis is a detailed and comprehensive analysis of optical burst switched networks, which is presented in Chapter 2. A shorter version of this analysis has been published as an entry in the *Encyclopedia of Multimedia Technology and Networking* by Idea Group Reference [17].

The second contribution of this thesis is the proposal, implementation and validation of an object-oriented model for simulation of IP over optical burst switching mesh networks, which is described in Chapter 3. This model was presented at the *10th IEEE Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD'2004),* as part of the *IEEE Global Telecommunications Conference (GLOBECOM'2004)* [18]. An extended version of this paper has been accepted for publication as a chapter of the book *Simulation and Modeling: Current Technologies and Applications* [19], published by Idea Group, Inc.

The third contribution is the proposal of two new performance metrics for performance assessment of OBS networks. These performance metrics, the nodal degree gain and the chord length gain (which only applies to chordal ring topologies) are described in Section 2 of Chapter 4. The chord length gain was reported in [20]. Regarding nodal degree gain, a special case only valid for OBS networks with degree-three chordal ring topologies, has reported in the *Kluwer Telecommunications System Journal* [20], which was further extended to degree-four chordal rings in [21-24]. A generalization of nodal degree gain valid for OBS networks with mesh topologies has been presented in [25, 26].

The fourth contribution is a performance study of OBS networks with ring and degree-three chordal ring topologies for JIT, JumpStart, JIT$^+$, JET, and Horizon protocols. This study is presented in Section 3 of Chapter 4. A first performance study, considering only JIT and JET protocols, was presented at the *3rd International Conference on Networking* [21]. This paper was considered as one of the best papers among 169 accepted papers out from 320 submissions and was selected for publication, as an extended paper, in the *Kluwer Telecommunications System*

Journal [20]. This extended version of [21] took into account the five one-way resource reservation protocols: JIT, JumpStart, JIT$^+$, JET, and Horizon.

The fifth contribution is the study of the performance of OBS networks with degree-four chordal ring topologies for the five one-way resource reservation protocols under study. This study, reported in Section 4 of Chapter 4, was presented at the *12th International Conference on Telecommunications* [22].

The sixth contribution is the performance assessment of the five considered one-way resource reservation protocols in optical burst switching networks with mesh topologies of nodal degrees up to four. A preliminary version of this work, considering only mesh topologies with nodal degrees up to three, has been presented at the *International Conference on Information Networking (ICOIN 2004)* [23]. This paper was further selected for publication, after revision, in the book *Information Networking: Networking Technologies for Broadband and Mobile Networks* included in the *Lecture Notes in Computer Science* book series of Springer-Verlag [24]. A further study, for topologies with nodal degrees up to four, has been presented at *7th International Conference on High-Speed Networks and Multimedia Communications* [27].

The seventh contribution is a proposal of the OBS network topology for the backbone of the Potuguese *Fundação para a Computação Científica Nacional* (FCCN) and its performance evaluation. This study has been presented, in Portuguese, in the Posters Session at the *7ª Conferência sobre Redes de Computadores (CRC'2004)* [28]. A further study, comparing this proposal with degree-three chordal ring topology with same number of nodes, has been presented at the *5th Conference on Telecommunications* [25].

The eighth contribution is a detailed study of the impact of the nodal degree on the performance of optical burst switching networks. Preliminary versions of this study have been presented at the *International Conference on Information Networking* [29] and at the *4th International Conference on Networking* [30], for networks with 16 and 20 nodes, respectively. The former was one of the 22% of accepted papers out of 427 submissions and published in the *Lecture Notes in Computer Science* book series of Springer-Verlag. The latter was considered as one of the best papers among 238 accepted papers out of 651 submissions and was selected for publication, as an extended paper, in the *Springer Telecommunications System Journal* [166]. Continuing this study, the influence of the nodal degree on nodal degree gain (considering mesh topologies with nodal degree between three and six)

was presented at the *International Conference on Systems Communications 2005* [26]. This conference accepted 80 papers out of more than 200 submitted.

The nineth contribution is a study of the influence of the setup message processing and the optical switch configuration times on the performance of optical burst switched mesh networks. A preliminary version of this study was presented at the *20th International Symposium on Computer and Information Sciences* [32]. This conference accepted approximately 30% of 491 submissions, this paper being one of the 20% accepted for publication in LNCS by Springer-Verlag, while the remaining 10% were selected for publication in a special edition of "Advances in Computer Science and Engineering Series" by Imperial College Press.

The tenth contribution is the formal specification and performance assessment of a new one-way resource reservation protocol, named Enhanced JIT (E-JIT). This contribution was published as a technical report [33] and it will be further submitted for publication in a refereed international journal.

Chapter 2

Optical Burst Switched Networks

2.1 Introduction

The concept of burst switching was initially proposed in the context of voice communications by Haselton [167] and Amstutz [168, 169] in the early 1980s. In 1990, Mills *et al*. [170] referred to burst switching as a principle presented within the Highball project. More recently, in the late 1990s, optical burst switching (OBS) was proposed as a new switching paradigm for optical transport networks [1, 54, 55, 57, 63-65, 73]. As Perros observed in [171], OBS is similar to an adaptation of a standard of International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) for burst switching in Asynchronous Transfer Mode (ATM), known as ATM block transfer (ABT), considering ABT with immediate transmission. In ABT with delayed transmission, each cell is stored and forwarded individually by the switch [62].

There is no unique definition of OBS in the literature. Nevertheless, some widely agreed common characteristics for OBS have been presented by Dolzer *et al*. [4]; as follows:

- Granularity — the transmission unit size (burst) of OBS is between the optical circuit switching and optical packet switching;

- Separation between control and data — control information (header) and data are transmitted on different channels (or wavelengths) with some time interval;
- Allocation of resources — resources are allocated without using explicit two-way source to destination signaling. A source node does not need to wait for the acknowledge message from the destination node to start transmitting the burst;
- Variable burst length — the burst size is variable;
- No optical buffering — burst switching does not require optical buffering in the intermediate nodes (without any delay).

OBS [47, 54, 57, 58, 63, 64, 74, 75] aims to overcome technical limitations of the optical packet switching, namely the lack of optical random access memory and the problems with synchronization. OBS is a technical compromise between wavelength routing (circuit switching) and optical packet switching, since it does not require optical buffering or packet-level processing as in optical packet switching and it is more efficient than circuit switching if the traffic volume does not require a full wavelength channel.

An approach to improve the performance of OBS networks was proposed by Verma *et al*. [5]. They presented a traffic shaping scheme to set the offset between the successive data bursts of a given data stream (label switch path), and their associated control packets, which randomizes the offset value to reduce the burst loss probability. This proposal considers the use of MPLS on OBS networks.

Recently, Qiao [110, 172] also proposed a solution that extends the generalized multiprotocol label switching (GMPLS) to OBS, called labeled optical burst switching (LOBS). LOBS is similar to multiprotocol lambda switching (MPλS) [77-79] proposed for wavelength-routed networks. LOBS is characterized by having control packets that contain labels and follow a pre-established label switched path, while the data are sent in bursts following their corresponding control packets as in OBS. When this mechanism is used, it is necessary to increase each OBS node with an IP/GMPLS controller, and thus it becomes as LOBS node, which is similar to a label-switched router. Therefore, in LOBS, control packets carry additional label information. LOBS layer provides OBS services namely, burst assembly, WDM topology and resource dissemination, and survivability. This framework is similar to GMPLS in that different LOBS paths can share the same λ (lambda). It differs from GMPLS because LOBS has the following exclusive issues: assembly (offset time), quality of

service[8] (QoS) in bufferlless core network, routing and λ-assignment, contention resolution,
light-spitting (for WDM multicast).

In [173], a proposal is made for a new service differentiation architecture and control plane scheduling mechanisms for providing QoS guarantees in LOBS networks. Service differentiation is achieved by a combined use of burst assembly algorithm and a fair packet queuing (FPQ) scheduler. The burst assembly algorithm controls the blocking probability and assembly delay using two parameters: the maximum burst size and the assembly timeout. A FPQ scheduler is used in edge nodes to regulate the access to a wavelength reservation algorithm.

Another variant of traditional OBS was proposed by Detti *et al.* [174], called optical composite burst switching (OCBS). This technique is based on two main features: (1) IP packets are assembled into bursts and (2) the burst contention in an optical switch is handled by means of two techniques — the wavelength dimension and the burst-dropping technique. Under OCBS the switch adopts the burst-dropping technique, which discards only the initial part of the burst, finding all of the engaged output wavelengths while forwarding the rest of the burst, beginning at the time that one wavelength becomes free. These authors illustrated and analyzed the burst-dropping technique and concluded that it allows an increase in the performance of the switch, in terms of packets loss probability, considering that only the wavelength dimension technique is adopted.

This thesis does not consider the above-mentioned methods, focusing only on OBS networks. This chapter presents an overview of OBS networks. The remaining part of this chapter is organized as follows. Section 2.2 is dedicated to OBS network architecture, which includes the description of edge and core nodes. Section 2.3 discusses the burst assembly process and algorithms. Section 2.4 addresses OBS resource reservation protocols, taking into account the way reservation and focusing on one-way OBS resource reservation protocols. In this section, a new resource reservation protocol, called Enhanced JIT (E-JIT), is proposed. Section 2.5 focuses on the problem of contention and other OBS issues like quality of service, TCP over OBS, Multicasting, burst grooming, and OBS applications are presented in Section 2.6. Main conclusions are presented in Section 2.7.

[8] Quality of service (QoS) in OBS networks will be addressed in Sub-section 2.6.1. However, in each section the respective mechanisms that include QoS issues are mentioned.

Sections 2.2 and 2.3 are partially based on the paper [17] and Section 2.4 is partially based on the papers [17, 27, 32, 166].

2.2 OBS Network Architecture

Network architecture defines the structure and the behavior of the real sub-system that is visible for other interconnected systems, while they are involved in the processing and transfer of information sets [87]. The OBS network architecture [175] follows the Open Systems Interconnection (OSI) reference model from International Organization for Standardization (ISO) [88] and the Transmission Control Protocol/Internet Protocol (TCP/IP) architecture [176-179]. Figure 2.1 shows a comparison of the different layers of these three network architectures [87, 175]. The OBS layer supports a set of tasks, namely, physical layer services and functions, and OBS connection management services [175]. Thus, the OBS network architecture implements the lower two layers, namely, physical and data link layer. Some examples of OBS connection management services and physical layer services and functions are the following: signaling, transmission and reception of optical bursts, Opto-Electronic conversion of control packets, burst framing; generation of offsets, emission burst control packets, wavelength multiplexing and demultiplexing, wavelength assignment, and resource reservations.

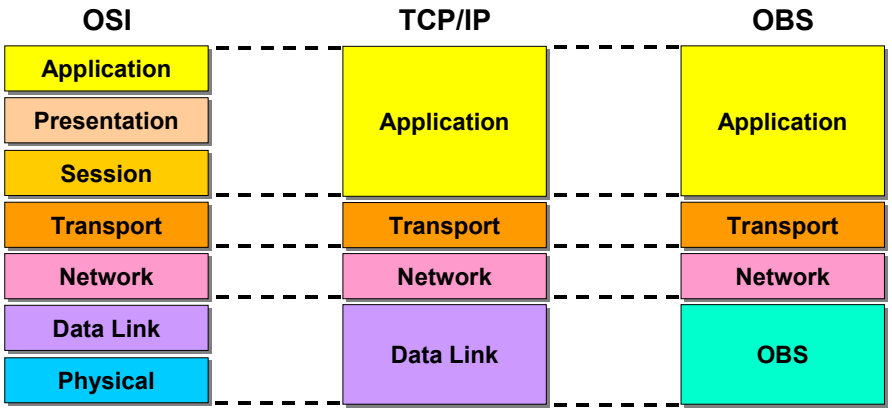


Fig. 2.1. Generic Network Architecture Layers: Comparison of OSI, TCP/IP, and OBS models.

In OBS networks, IP packets (datagrams) are assembled into very large size packets called data bursts. These bursts are transmitted after a burst header packet (also called setup message or control packet), with a delay of some offset time in a

different data channel. Burst offset is the interval of time, at the source node, between the processing of the first bit of the setup message and the transmission of the first bit of the data burst. Each control packet contains routing and scheduling information and is processed at the electronic level, before the arrival of the corresponding data burst [5, 57, 75, 180]. The transmission of control packets forms a control network (a packet-switch network) that controls the routing of data bursts in the optical network [95]. In the OBS domain, there are two types of channels (or wavelengths) [181]: the control channels that carry the burst control packets (*Setup* messages), called "control channel group" (CCG); and the data burst channels that carry the data bursts, called "data burst channel group" (BCG). Thus, two channels are assigned and scheduled for transmission of the burst control packet and its correspondent data burst. Figure 2.2 illustrates a schematic representation of a burst with the respective setup message and offset time. In this figure, *Channel 1* belongs to CCG and *Channel 2* belongs to BCG.

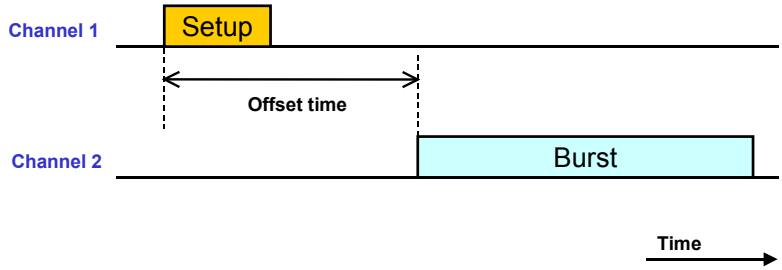


Fig. 2.2. An example of OBS.

An OBS network is an all-optical network where core nodes (optical cross connect – OXC, plus signaling engine or switch control unit - SCU) transport data from/to edge nodes (IP routers) that are interconnected by bi-directional links, as may be seen in Figure 2.3. Edge and core nodes are presented in Sub-sections 2.2.1 and 2.2.2, respectively.

Figure 2.3 presents an example of an OBS connection where input traffic (packets) comes from source edge *Node A* to destination edge *Node B*. The source edge node is referred to as the ingress node and the destination edge node is referred to as the egress node. The ingress node of the network collects the upper traffic layer (various types of client/user data), sorts and schedules it into electronic input buffers based on each packets' class and destination address. These packets are aggregated into bursts and are stored in the output buffer, where electronic Random Access Memory (RAM) is cheap and abundant [63]. After the burst assembly

process, a control packet is created and immediately sent towards the destination to set-up a connection for its corresponding burst. After the offset time, the burst is transmitted all-optically over OBS core nodes without any storage at the intermediate nodes within the core, until the destination edge node. After receiving a burst, the egress node disassemblies it into packets and provides these packets to the upper layer. The packets are forwarded electronically to destination users (output traffic) [60, 90-93].

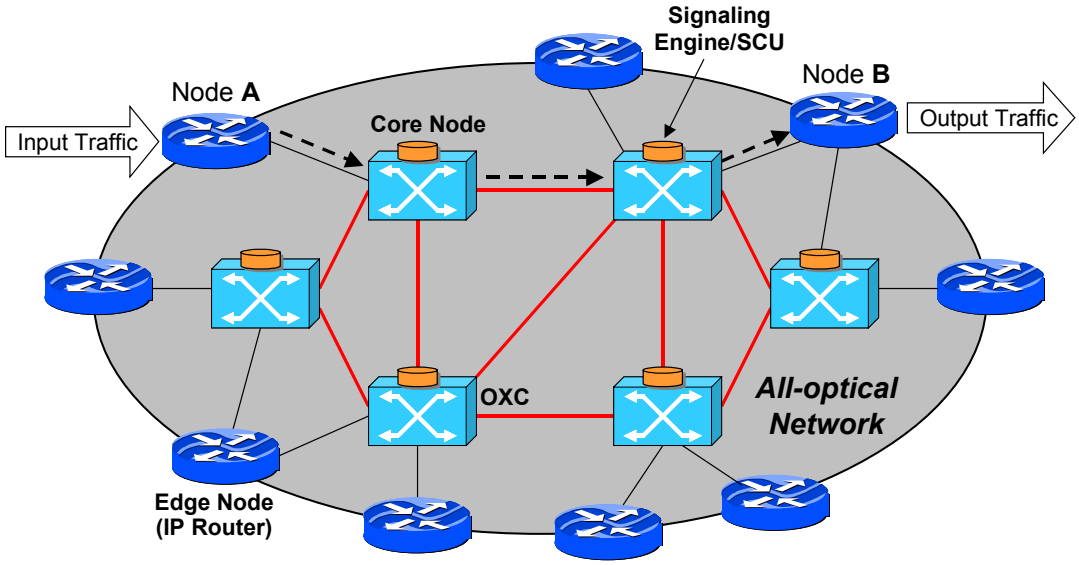


Fig. 2.3. Schematic representation of an IP over OBS network.

Düser and Bayvel [114] propose an alternative OBS network architecture, called wavelength-routed optical burst switching (WR-OBS), that combines OBS with fast circuit switching which uses an end-to-end resource reservation procedure. This scheme uses a centralized request server (centralized signaling) that is responsible for resource scheduling of the entire OBS network. At the edge nodes, all processing and buffering (including, the burst assembly) are concentrated and bursts are sent through the core (backbone) network using dynamic wavelength allocation. It is assumed that there are neither buffers nor wavelength conversion in the core nodes.

2.2.1 Edge Node

The OBS edge node works as an interface between a typical IP router and the OBS backbone [91, 94]. As mentioned above, the behavior of an edge node when it establishes a connection was described. Thus, Kan *et al*. [91] summarized the following operations that an OBS edge node needs to perform:

- Assemble IP packets into data bursts based on some assembly policy;
- Generate and schedule the control packet for each burst;
- Convert the traffic destined to OBS network from electronic signal to optical domain and multiplex them into WDM wavelength;
- Demultiplex incoming wavelengths channels and perform optical-to-electronic conversion on the traffic received;
- Disassemble and forward IP packets to the regular routers connected.

Edge nodes need to provide several interfaces such as Gigabit Ethernet, ATM, and IP [182]. The architecture of the edge node consists of three modules [90, 93]: routing module, burst assembly module, and scheduler module. Figure 2.4 shows the schematic representation of the architecture of an edge node [63, 90, 93]. The routing module selects the appropriate output port for each packet and sends it to the correspondent burst assembly module. The latter assembles bursts containing packets that are addressed for a specific egress node. In this module, there is a different packet queue for each class of traffic. There are usually different assembly queues for each class of traffic (or priority). The burst scheduler module creates a burst and their corresponding control packet based on the burst assembly policy, and sends it to the output port.

In OBS networks, the most important task developed by the edge node is the burst assembly process. This issue will be analyzed in detail in Section 2.3.

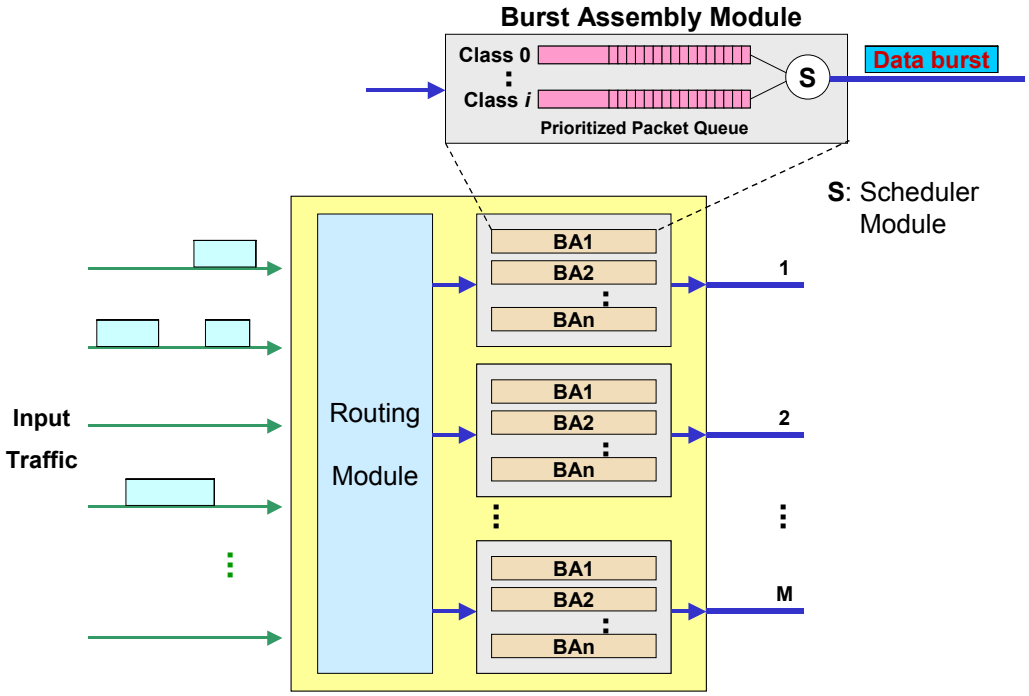


Fig.2.4. Schematic representation of the architecture of an OBS edge node.

2.2.2 Core Node

The OBS core node consists of two main components [90, 93, 95]: an optical cross-connect (OXC) and a switch control unit (SCU) or signaling engine. The SCU implements the OBS signaling protocol. It also creates and maintains the forwarding table and configures the OXC. It is assumed that SCU is implemented in hardware to avoid the bottlenecks in the control plane and to accomplish operations at high speeds [80]. An example of this unit is JITPAC (Just-in-Time Protocol Acceleration Circuit) hardware [141], developed to implement the just-in-time (JIT) resource reservation protocol in the scope of JumpStart project [86].

An OXC is an *N* x *N* optical switch, with *N* input fibers and *N* output fibers. OXC performs its optical switch matrix to conduct the data burst to the next node. Several optical switch matrices were developed, namely the broadcast-and-select proposed in [96] and the well-known switching fabrics presented in [97-99]. In this thesis only OXC with non-blocking space-division switching fabrics with no optical buffers are considered. Figure 2.5 shows the operation of such an optical burst switching core node [37, 172]. Data channels are connected to optical switch fabric and control channels are terminated at SCU. Between each control packet and the

correspondent data burst there is an offset time (*T*) to compensate for the processing/configuration delay [63]. The SCU function is similar to a conventional electronic router [95].

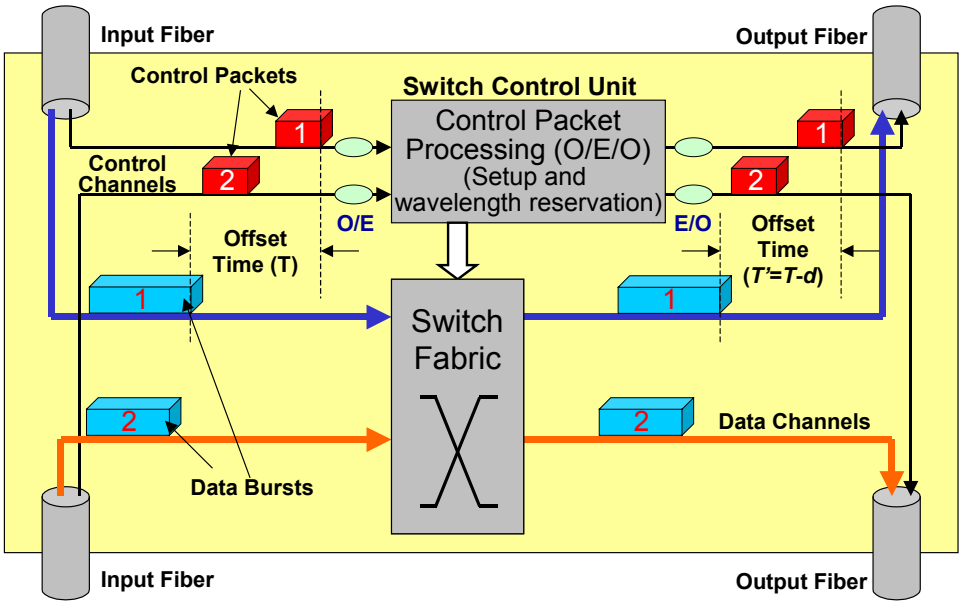


Fig. 2.5. Schematic representation of the architecture of an OBS core node.

When SCU receives a control packet, it converts the packet from optical to electronic signal, processes the setup message, identifies the intended destination, and consults the forwarding table to find the intended output port. If the output port is available in the instant that the data burst arrives, SCU configures the switch fabric to let the burst pass through. If the port is not available, depending on the contention resolution implemented in the network, either (1) the arrival burst is dropped or (2) the current burst that is occupying the port is segmented or dropped. On the other hand, if data burst arrives to OXC before the correspondent control packet, the burst is dropped. After a data channel reservation, SCU re-generates the control packet, converts the electronic signal into optical domain, and sends it to the next node [90, 95, 101]. As it can be seen in Figure 2.5, the new offset time (*T'*) for next hop is equal to *T*-*d*, where *d* represents time delay for processing the control packet and configuring the switch fabric.

Kan *et al.* [100] summarize the operations that an OBS core node needs to perform, which are the following:

- Demultiplex wavelength data channels;

- Terminate data burst channels and conduct wavelength conversion for passing through optical switch fabric;
- Terminate control packets channels and convert control information from optical to electronic domain;
- Schedule incoming bursts, send instructions to the optical switch matrix, and switch data burst channels through the optical switch matrix;
- Re-generate new control packet for outgoing bursts;
- Multiplex outgoing control packets and bursts together into single or multiple fibers.

Based on the most important tasks performed by an OBS core node, Sections 2.4 and 2.5 focus on resources reservation and contention resolution issues.

In 2003, Aldwairi *et al*. [183] presented a new switch architecture for optical burst switched networks. This proposal was designed and implemented to demonstrate the just-in-time protocol. This switch is capable of processing 2.88 Gbps of signaling messages traffic and it was developed in the advanced technology demonstration network (ATDNet) [141].

2.3 Burst Assembly

2.3.1 Burst Assembly Process

Burst assembly [64, 90, 93, 95, 102-104], in OBS networks, is basically the process of aggregating and assembling packets into bursts at the ingress edge node. At ingress OBS node, packets that are destined for the same egress OBS node and belong to the same Quality of Service (QoS) class are aggregated and sent in discrete bursts, with times determined by the burst assembly policy. Burst assembly process is made into burst assembly module, inside edge node (Figure 2.6). Packets that are destined to different egress nodes go through different assembly queues to burst assembly module. In this module, the burst control packet is generated and correspondent data burst is assembled. At the egress OBS node, this burst is subsequently de-aggregated and forwarded electronically.

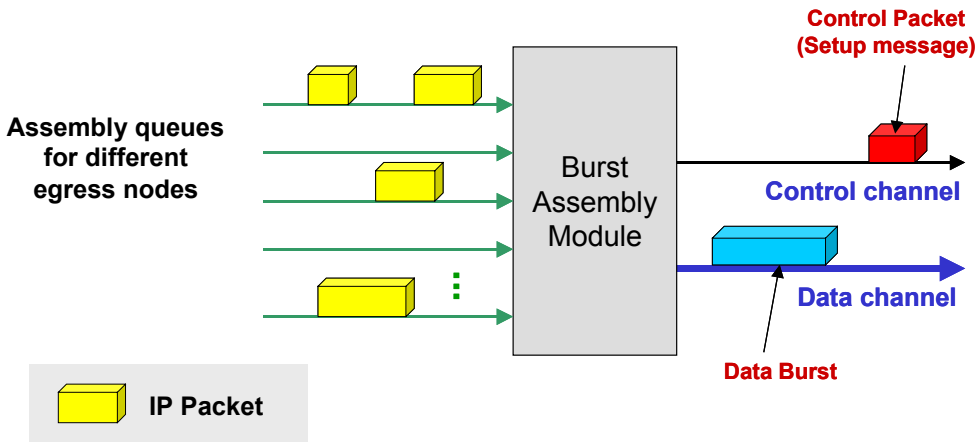


Fig. 2.6. Burst assembly process.

In the burst assembly process there are two parameters that determine how the packets are aggregated: maximum waiting time (timer value) and minimum size of the burst (threshold value) [184]. Based on these parameters, several burst assembly algorithms were proposed. These algorithms are addressed below in Sub-section 2.3.2.

2.3.2 Burst Assembly Algorithms

Based on the above-mentioned parameters that determine how packets are aggregated, burst assembly algorithms may be classified into the following three main categories:

- Timer-based algorithm [102, 104];
- Threshold-based algorithm or burst length-based algorithm [90];
- Hybrid algorithm or mixed timer/threshold-based algorithm [63, 95, 105].

In timer-based algorithms, the timer is started at the initialization of the system and immediately after the previous burst is sent. When the timer expires, the burst assembler generates a burst with all the packets in the buffer at that point [102]. This algorithm has the advantage of under low input traffic guaranteeing minimum delay for burst assembly, while under high input traffic, it may generate bursts that are quite large, possibly unnecessarily increasing delay.

An extension of timer-based technique, called assured Horizon, has been proposed by Dolzer [106, 107]. The basic idea of assured Horizon is the introduction

of a coarse-grained (or static) bandwidth reservation for every forwarding equivalent class between ingress and egress nodes.

Cao *et al*. [104] propose three timer-based algorithms and evaluate their performance in terms of effective data that the destinations receive, and burst loss rate. The Fixed-Assembly-Period (FAP) algorithm is a timer-based scheme. The Adaptive-Assembly-Period (AAP) algorithm is similar to FAP, with the difference that AAP can dynamically change the value of the assembly period (timer) of any queue at every ingress node according to the length of the burst recently sent. The third algorithm is the Min-BurstLength-Max-Assembly-Period (MBMAP) that is configured in function of parameters maximum assembly period (MAP) and minimum burst length (MBL). This algorithm generates a control packet when a burst exceeds a MBL or when the assembly period times out. The authors conclude that AAP is the algorithm with best performance because it adapts assembly periods to match with the transfer control protocol (TCP) congestion control mechanisms.

Under threshold-based algorithm [90], the burst is assembled when the number of packets in the buffer arrives at a threshold value. The threshold specifies the number of packets to be aggregated into a burst. Using this assembly policy, all bursts have the same number of packets when entering into core network. An advantage of this scheme is that, under high input traffic, the threshold value will be reached quickly, minimizing any delay. On the other hand, under low input traffic, it may need to wait for a long time until the buffer reaches a threshold value.

In an hybrid solution [95, 105, 108], the two algorithms are applied concurrently. Both a timer and a threshold parameter are set and the burst is sent when the timer expires or the threshold value is reached. Both timer and threshold values can vary in function of the input traffic load. This addition is called adaptive hybrid burst assembly [92].

Another hybrid solution proposed by Qiao [110], is the hybrid burst priority (HBP) scheme, integrated in LOBS framework, although it can also be applied to OBS and similar networks. Under HBP, packets with different priorities may be aggregated in the same burst. The priority of the burst is calculated as the weight average of priorities of each byte of the burst, rounded to the nearest integer. Packets with different priorities are ordered in the burst using the NutShell packet ordering scheme. Following this scheme, considering packets of classes 1, 2, …, 8, the packets with low priority (class 1) are put at the very beginning and/or end of the burst, then packets of class 2 as close to the two ends as possible and so on, in order to centralize the higher priority packets as much as possible. The name of this

scheme – NutShell – comes from the analogy of the protection the highest priority packets (as a nut) with lower priority ones (as a shell) at each side.

Tachibana *et al*. [109] proposed an approach based on round-robin burst assembly and constant burst transmission with scheduler. This method considers that burst assembly is made in the ingress edge node by three elements: a burstifier, a scheduler, and a burst switch. The burstifier has several buffers and packets arriving at the ingress node are stored in different buffers depending on their egress edge nodes destination. In each buffer, burst assembly is processed in round-robin manner and the cycle-time of round-robin can be defined as the total processing time at all buffers. Then, the burst assembly processing time is constant. The scheduler sends the control packet before the correspondent data burst with some offset time and they are transmitted to the burst switch at fixed intervals. In the burst switch, if there are no available output wavelengths, the control packet cannot reserve a wavelength and the correspondent burst is lost.

Recently, a mechanism to provide quality of service (QoS) support that consider bursts containing a combination of packets with different classes, called composite burst assembly was proposed [93, 103]. This mechanism makes good use of burst segmentation (technique used for contention resolution in the core network), where packets toward the tail of the burst have a higher probability of being dropped than packets at the head of the burst. Packet classes with low loss tolerance can be placed toward the head of the burst and packet classes with high loss tolerance can be placed toward the tail of the burst. The authors concluded that approaches with composite bursts perform better than approaches with single-class bursts in terms of loss probability and delay providing differentiated services for different class of packets (with QoS requirements).

In [184], the authors evaluated the performance of timer and threshold based burst assembly algorithms. They compared the results obtained by simulation with two theoretical models, the classical Engset model and an analytical model. They found that the burst blocking probability is not influenced by threshold based algorithms. However, for low load it presents lower blocking probability while under high load, the blocking probability is higher. With small bursts the performance is better in terms of blocking probability and link utilization for all range of load. Nevertheless, the exception was verified under low load for timer based algorithms and for threshold based algorithms although its effect is very weak.

2.4 Resource Reservation Protocols for OBS Networks

In order to transmit a burst over the OBS network, a resource reservation protocol must be implemented to allocate resources and configure optical switches for that burst at each node [64]. OBS signaling is usually implemented using out-of-band control packets. Resource reservation proposed in research literature can be classified in function of the following characteristics: the way of reservation (one-way reservation or two-way reservation); immediate or delayed reservation; implicit or explicit release; and centralized or distributed control protocols. In this thesis, resource reservation protocols are presented following the way of reservation criteria. Taking into account both categories (one-way and two-way reservation), this section focuses on one-way resource reservation protocols considering the most relevant protocols for OBS networks today.

At each OBS core node, after configuration of the optical switch for an incoming burst, it is necessary to reserve an outgoing data channel for that burst. Therefore, another important issue in OBS networks is the data channel scheduling. Taking into account that the role of signaling and channel reservation protocols are performed at each intermediate core node, this section focuses on resources reservation schemes together, mainly, in one-way resource reservation protocols.

2.4.1 Classification of Resource Reservation Protocols

According to the way of reservation, resource reservation protocols may be classified into two classes: one-way reservation and two-way reservation. In the first class, a burst is sent shortly after the setup message, and the source node does not wait for the acknowledgement sent by the destination node. Therefore, the size of the offset is between transmission time of the setup message and the round-trip delay of the setup message, reducing the end-to-end data transfer latency. Different optical burst switching mechanisms may choose different offset values within this range. Tell And Go (TAG) [111], Just-In-Time (JIT) [74, 75], JumpStart [75, 82-86], Horizon [1, 54], Just-Enough-Time (JET) [57, 73, 81], and JIT^{+} [80] are examples of resource reservation protocols using one-way reservation.

In the TAG protocol, a source node sends a control packet and immediately after sends a burst. At each intermediate node, the data burst has to go through with an input delay equal to the setup message (control packet) processing time. If a channel cannot be reserved on a link, along the ingress-egress path, the node preceding the blocked channel discards the burst. To release the connection, a "tear-down" control signal or packet is sent [111, 112]. In this protocol a burst may need to be delayed (buffered) at each node, while it waits for the processing of setup message and the configuration of the OXC switch fabric. TAG is practical only if the switch processing time of the setup message and the optical switch configuration time are very short [113]. Due to this critical limitation of TAG, this study concentrates on the study of the other above-mentioned resource reservation protocols that will be described in Section 2.4.2.

The offset in two-way reservation class is the interval of time between the transmission of the setup message and the reception of the acknowledgement from the destination. The major drawback of this class is the long offset time, which causes a long data delay. Examples of signaling protocols using this class include the Tell And Wait (TAW) protocol [111] and the Wavelength Routed OBS network (WR-OBS) proposed in [114].

In TAW protocol the source node sends a request packet (setup control message) along the path to the egress node informing that it wants to transmit a burst. If all intermediate nodes accommodate, the request is accepted and the source transmits. Otherwise, the request is refused, and the source node tries again. At an intermediate node, when it receives the setup message, the switch control unit reserves a free data channel on the routed output; such data channel is dedicated to the burst until an explicit release message is received. The egress edge node sends a *confirm* message in reverse direction with the aim of notifying the ingress edge node of the success of the setup phase of the optical virtual path. This confirmation message arrives at ingress edge node if a free data channel was found on each link. Every intermediate node at the virtual path will forward the confirmation message only if the configuration of the optical switching fabric is terminated, otherwise, the forwarding of the *confirm* message is delayed. When the ingress edge node receives the *confirm* message, it transmits the burst on the reserved optical virtual path. As soon as the transmission of the burst is over, the source node transmits a release message, which will "tear-down" the optical virtual path and free the data channels reserved along the path [111, 112].

The two-way reservation scheme proposed by Düser and Bayvel [114] (presented above in Section 2.2), called Wavelength-Routed Optical Burst Switching (WR-OBS), combines OBS with fast circuit switching. WR-OBS requires an obligatory end-to-end acknowledgement and it intends to provide a scalable architecture and quality of service (QoS) guarantees. After burst assembly process, an end-to-end data channel is requested for transmission the burst from the source to the destination edge node. Once a free data channel is found, the burst is assigned to it and is transmitted to the core network. After burst transmission, the data channel is released and can be reused for new connections.

Due to the impairments two-way reservation class and the critical limitation of TAG, the study focuses on one-way reservation schemes, being considered, hereafter, the following resource reservation protocols: JIT, JumpStart, JIT$^+$, JET, and Horizon. Chapters 4 and 5 will focus on performance assessment of these one-way resource reservation protocols for optical burst switched networks.

2.4.2 One-way OBS Resource Reservation Protocols

Most of the proposed OBS architectures use one-way resource reservation protocols to transmit bursts through the network [60]. Qiao and Yoo [57] consider that an one-way reservation scheme is appropriate because OBS is a more suitable technology to be implemented in long-haul networks. Thus, it will significantly decrease the time needed to establish a connection. Wei and McFarland [74], who proposed JIT protocol, analyzed the setup latency of JIT and compared it with circuit switching. They concluded that one-way OBS signaling scheme has a much shorter setup time and better throughput performance.

Several one-way OBS resource reservation protocols were proposed for optical burst switched networks. As mentioned before, this thesis focuses on the most relevant protocols based on one-way reservation presented on research literature, namely, Just-in-Time (JIT) [74, 75], JumpStart [75, 82-86], JIT$^+$ [80], Just-Enough-Time (JET) [57, 73, 81], and Horizon [1, 54] resource reservation protocols.

Research literature does not agree about Horizon protocol, proposed by Turner in Terabit Burst Switching [1, 54, 185], in terms of signaling and/or scheduling functionalities. Horizon is mentioned only as a scheduling algorithm by researchers

from the Computer Science and Engineering Department, at the State University of New York at Bufalo [63] and from Department of Computer Science of the University of Texas at Dallas [64]. However, in [60, 62, 75, 80, 85, 115, 133, 137] and [4, 107, 186], it is presented by researchers from the Department of Computer Science of the North Carolina State University and from Institute of Communication Networks and Computer Engineering, University of Stuttgart, respectively, as a resource reservation scheme. In this study, this second approach is followed and Horizon is considered a resource reservation protocol in the sense that it performs a delayed reservation as mentioned in [80, 133] and as it is described in sub-section 2.4.2.3.

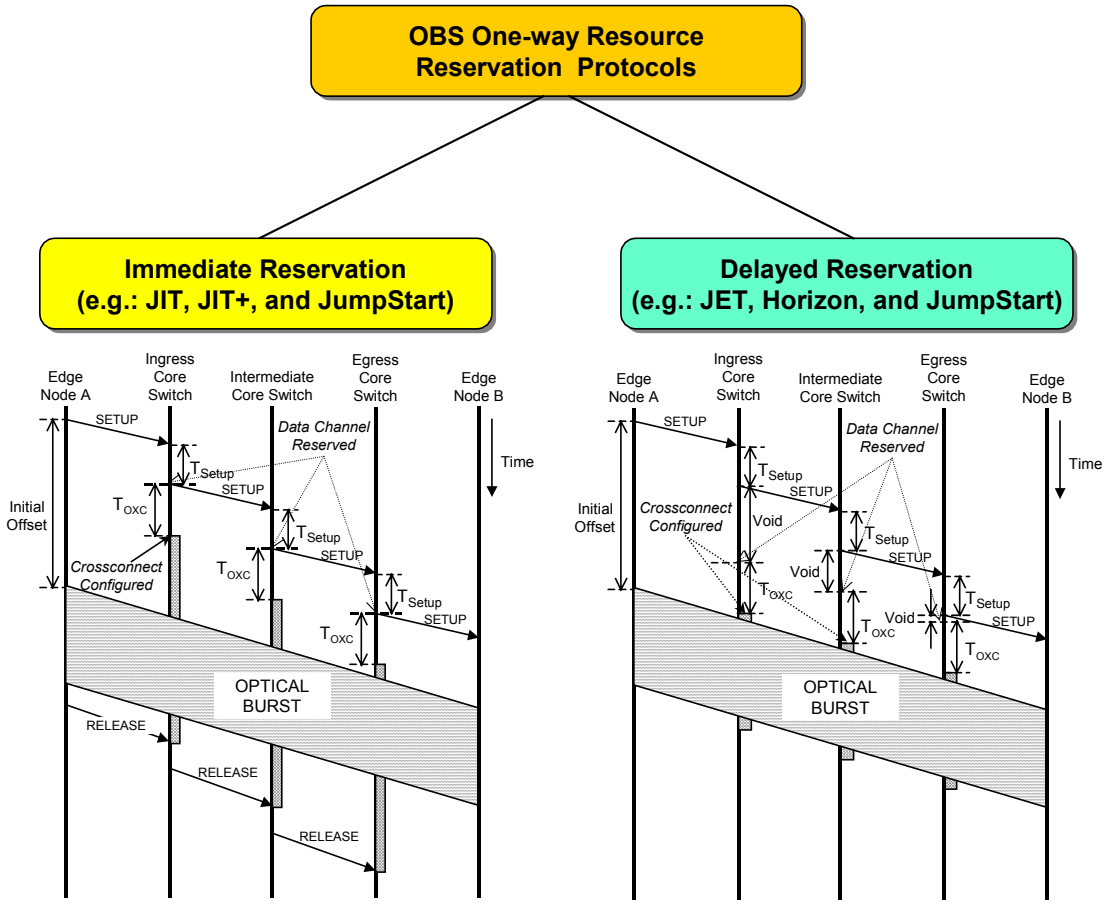


Fig. 2.7. Classification and message flow of one-way resource reservation protocols for OBS networks.

Figure 2.7 shows that one-way resource reservation protocols may be classified, regarding the way in which output data channels are reserved for bursts, as immediate and delayed reservation. In immediate reservation schemes, an output channel of the optical cross-connect (OXC) is reserved for the incoming bursts immediately after the arrival of the corresponding setup message. In delayed

reservation, the resources of the OXC are allocated for a burst just before the arrival of the first bit of the burst. JIT and JIT$^+$ resource reservation protocols are examples of immediate wavelength reservation, while JET and Horizon are examples of delayed reservation schemes. The JumpStart signaling protocol may be implemented using either immediate or delayed reservation [62].

Using an immediate reservation protocol, OXC elements are not reserved for future bursts and, therefore, no scheduling takes place in an OBS node. This results that OXC matrix is very simple [115]. Otherwise, channel scheduling would be more complicated. Usually, immediate reservation protocols use a first-come, first-served scheduling, i.e., bursts are switched in the same order in which their corresponding setup messages arrive at the node.

Figure 2.7 shows the following time parameters that play an important role in the study of the performance analysis in OBS networks:

- $T_{Setup}(X)$ represents the amount of time that is needed to process the setup message in an OBS node according to a format defined by the resource reservation protocol X; X can be JIT, JumpStart, JIT$^+$, JET, or Horizon. The time is not necessarily the same for every protocol, but it is assumed that it is equal in every node in the network, since they all function under the same protocol.

- T_{OXC} is the time an OXC needs to configure its switch fabric to establish a connection between an input port and an output port. This may also be viewed as the time the OXC takes, after interpreting the command in the control message, to position correctly the micro-mirrors (micro-electro-mechanical systems - MEMS - switch [163, 187]) in the matrix and to be used to switch a burst.

- $T_{Offset}(X)$ is the burst offset time for the resource reservation protocol X; X can be JIT, JumpStart, JIT$^+$, JET, or Horizon. The value of offset time depends on the resource reservation protocol and the number of nodes that the burst has already passed through (this value decreases as the burst crosses a node, because each node processes electronically the control message, delaying it). The main concern is to make this value such that the first bit of the burst arrives at the egress node shortly after the node is configured and ready to receive it. If k is the number of OBS nodes

in the path of a burst from source to destination, then the minimum value to $T_{Offset}(X)$ is [80]:

$$T_{offset}^{(\min.)}(X) = k.T_{Setup}(X) + T_{OXC} \quad (2.1)$$

The sub-sections that follow, present in detail each one of the one-way resource reservation protocols referred to above. The operation of different resource reservation protocols is illustrated in Figures 2.8-2.13 where the three-above mentioned parameters and the following two are considered:

- T_{idle} is the idle time in an output channel in the OBS node. During this period of time, the OXC is already configured and it is still waiting to switch a burst.
- T_{void} represents the time in which an output channel is still free and waiting to be configured for receiving (and switching) a burst.

In these figures (Figures 2.8, 2.11-2.15 and 2.18-2.20), it is considered a single data channel of an OBS node to present the operation of resource reservation protocols. A temporal line is used to illustrate the sequence of events. In this time line the different instants in which one can observe better the sequence of events to reserve a data channel to switch a burst are represented. However, the time line does not present each time in terms of the same scale due to space limitations (e.g. T_{OXC} e larger than T_{Setup}).

At the instant t_0, it is considered that the OXC receives the setup message (*Setup A*) and, between t_0 and t_1 converts it from optical to electronic signal to be processed. The instant t_1 is the minimum time that one may consider for the data channel w_1 of the OXC to be occupied.

In these sub-sections, T_{Setup} is used without referring to the resource reservation protocol *X*, since it is referred to each time in the context of the corresponding protocol.

2.4.2.1 Just-in-Time (JIT)

Just-in-Time (JIT) resource reservation protocol was proposed by Wei and McFarland in December 2000 [74]. It was developed under the Multiwavelength Optical Networking program (MONET) [188-190]. Under JIT, an output channel is reserved for a burst immediately after the arrival of the corresponding setup message. If a channel cannot be reserved immediately, then the setup message is rejected and the corresponding burst is lost.

JIT protocol is an example of one-way resource reservation protocol with immediate resources reservation. Figure 2.7 presents the operation of JIT protocol exemplifying immediate reservation schemes. As it may be seen in this figure, after processing the setup message (T_{Setup}), the data channel changes to *reserved* state immediately. As it may be seen, the T_{Offset} is set in a way that the sum of the successive T_{Setup} and T_{OXC} results in a minimum T_{Idle} at the egress switch.

In the connection example presented in Figure 2.7 (immediate reservation), data must cross three switches before it reaches its destination.

In general, T_{Offset} is as follows:

$$T_{Offset} \geq k.T_{Setup} + T_{OXC} \tag{2.2}$$

where k is the maximum number of hops the signal must jump to reach its destination in the network; to greater efficiency, the signal '$\geq$' must be understood as "slightly larger".

As it may be seen in Figure 2.7 (immediate reservation), T_{Idle} at the last node can be obtained as follows:

$$T_{Idle} = T_{Offset} - k.T_{Setup} + T_{OXC} \tag{2.3}$$

By (2.2) and (2.3) it is possible to see that T_{OXC} is considered only once. Irrespective of the relative greatness between T_{OXC} and T_{Setup}, the configuration of OXC_n (the n-th OXC of the network burst path) is only made after the configuration of OXC_{n-1} was started, and it is possible that they occur simultaneously. This is a consequence of both the nature of the signal and the hardware itself.

The various T_{Setup} are sequential and interdependent, that is, the intermediate OXC only sends the control message to the next hop after it electronically processes and converts the message to optical form. Thus, the total configuration time for the *k* nodes in the network is given by (2.2).

Figure 2.8 shows another perspective of the operation on JIT protocol, presenting two successive bursts (*Burst A* and *Burst B*) and the intention to receive the third. As it may be seen in this figure, at the instant t_2 (after T_{Setup}) it is assumed that the OXC processed the setup message of the *Burst A*. It configures immediately the switch fabric to switch the *Burst A* (at the instant t_3) and continue configured (T_{Idle}) waiting for it. The burst is switched between t_4 and t_5 (the burst length or burst size is equal to t_5 - t_4).

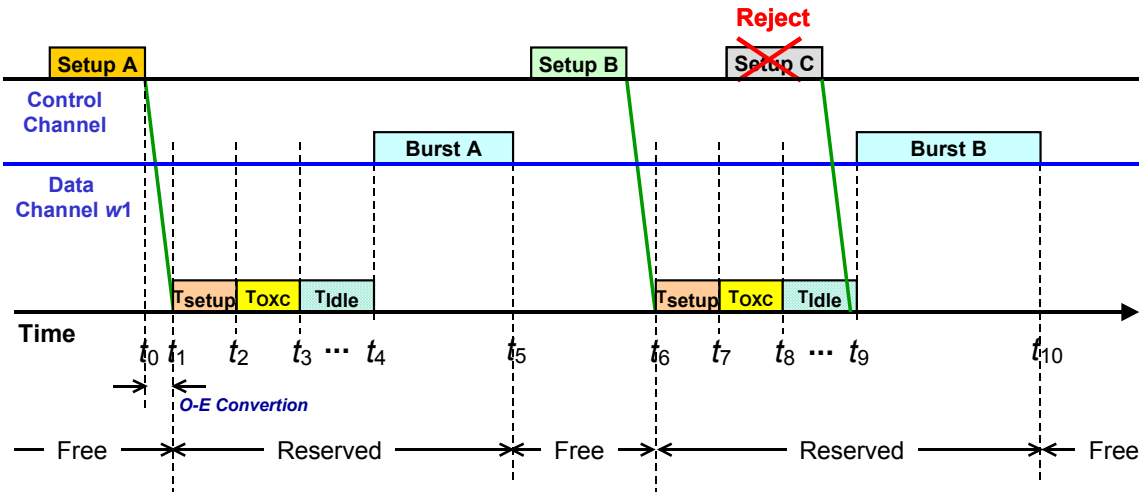


Fig. 2.8. Operation of JIT resource reservation protocol.

After time t_5 the OXC can accept a new burst (*Burst B*). *Burst B* works in the same way as *Burst A*. If the *Setup B* arrives at the switch at the instant $t<t_5$, it would be rejected and the correspondent burst will be lost.

Each data channel may have two states: “free” or “reserved”. If the wanted data channel is “free”, the setup message and the respective data burst are accepted, otherwise, the setup message is rejected and the correspondent data burst is lost. As it may be seen in Figure 2.8, even though the data channel w_1 may be *Idle* (between time t_8 and t_9), it is “reserved” for the *Burst B*. Thus, the setup message of the *Burst C* (*Setup C*) is rejected and the correspondent data burst is lost. After t_{10} (end of the *Burst B*) the state of the data channel changes to “free” again waiting for a new burst.

The change of the state to “reserved” takes place upon receipt of the setup message, whereas the change to state “free” occurs when the last bit of the burst arrives at the switch at time t_5 [80]. The length of state “reserved” is equal to the burst length plus the correspondent offset (T_{Offset}). The length of a state “free” is equal to the time until the arrival of the next setup message.

JIT protocol uses explicit releases to set free the switch fabric resources, although it is not represented the “release message” processing time in Figure 2.8 because it does not influence the resources reservation to switch a burst in terms of the description of protocol operation. This message is sent either by the source node or the destination node, to tear down all OXCs along the path on an existing connection trail. Whenever any network element detects a “setup” failure, it sends a “release message” to all network elements along the path to the source node. As mentioned above, it is assumed that the status of data channel w_1 is updated to “free” when the last bit of the burst arrives at the switch and it aims to present the operation of resources allocation using this protocol. Therefore, in this context, the representation of “release message” can be neglected.

This protocol uses first-come, first-served (FCFS) burst scheduling, in the sense that bursts are switched in the order in which their corresponding *setup messages* arrive at the node. An OBS node may choose the first free available data channel.

2.4.2.2 JumpStart

JumpStart [75, 82-85] is a joint project [86] supported by Advanced Research and Development Agency (ARDA) [191] developed by the North Carolina State University (NCSU) and MCNC Research and Development Institute [192]. The goal of JumpStart project is the definition of a signaling protocol and associated architecture for a WDM burst-switching network.

The authors consider the following basic architectural assumptions in the development of JumpStart architecture [84]:

- Signaling is done out of band (respecting the OBS paradigm);
- Data are transparent to intermediate network entities (i.e. no electro-optical conversion is done in intermediate nodes);
- There is a single high-capacity signaling channel/wavelength;

- Explicit setup - signaling messages are processed by all intermediate nodes and the cross-connect elements of each switch are configured for a burst immediately after the arrival of the associated setup message;
- Most network intelligence is concentrated on the edge; core nodes are kept simple;
- The implementation of the signaling protocol must be amenable to implementation in hardware;
- No global time synchronization between nodes is assumed.

In terms of signaling protocol, JumpStart design was guided by the following assumptions [75]:

- Signaling is out of band;
- Signaling channel is best-effort link by link (signaling protocol reliability is not desirable in a JIT OBS network because low error rate and increased burden on the signaling engine (or SCU) make the protocol reliable);
- Signaling messages are queued and processed by each intermediate node (queuing losses are possible);
- Signaling channel is presumed to possess a low bit error (e.g., 10^{-12} to 10^{-15}).

JumpStart is a JIT based resource reservation protocol. However, it has the following additional characteristics [75, 84, 85], performing:

- Both immediate (such as JIT) and delayed data channel reservation (such as Horizon);
- Both unicast (unicast asynchronous single bursts and lightpaths, and unicast bursts that belong to a persistent path connection) and multicast. Within multicast variety of connection there are two ways of starting a multicast session:
  - Source-managed multicast – the multicast source knows the address of all the members of the multicast group. The number of members is relatively small;
  - Leaf-initiated join – a source may announce the existence of a multicast session, with a *session id* that is unique inside the network;
- Persistent connections service (to both unicast and multicast sessions) – bursts that belong to the same logical connection and follow the same route through the network;

- Both short bursts and light-paths;
- Processing delay prediction – the setup message has to be sent far enough in advance before the correspondent burst;
- Quality of Service (QoS) – QoS in JumpStart is separated into several areas:
  - QoS requirements defined for the specific adaptation layer used;
  - Optical QoS parameters on which specific adaptation layer requirements may be mapped to;
  - Connection prioritization – it is possible to preempt less important connections in favor of more important ones;
- Label Switching – this is possible by adding properly formatted GMPLS labels to the signaling messages that establish new connections.

Under JumpStart [75, 115], a source edge OBS node first sends a setup message to its ingress OBS core node with information related to the burst transmission, including the source and destination addresses. If the ingress core node can switch the burst, it returns a *setup ack* message to the edge node. Moreover, it forwards the setup message to the next node. Otherwise, the ingress core node refuses the setup message and returns a *reject* message to the edge node and the corresponding burst is dropped. In this case, the edge node enters in an idle period waiting for another burst. When a new burst arrives, the edge node repeats the process. An example of the signaling messages operation exchanged between an edge node and its ingress core switch is presented in Figure 2.9.

A setup message can carry the burst length and a core node can use this information to free resources allocated to the burst. In this case, the node is said to employ *estimated release*. Otherwise, the source edge node needs to send a *release* message to indicate the end of the burst transmission (*explicit release*). The schematic representation of JumpStart signaling messages is showed in Figure 2.10, presenting a successful transmission of a burst and using immediate resources reservation. This figure uses the same notation of Figure 2.7 showed in sub-section 2.4.2.

After receiving the setup message (if no blocking occurs along the path), the destination edge node, can optionally send a “connect message” to the source edge node acknowledging the success of the transmission. If at each intermediate core node along the path between source and destination there is a switch that cannot switch the incoming burst, it sends a *reject* message to the previous node and it drops the burst. Upon receiving this *reject* message, previous nodes free their resources allocated to the burst and forward the message along the reverse path back to the source edge node.

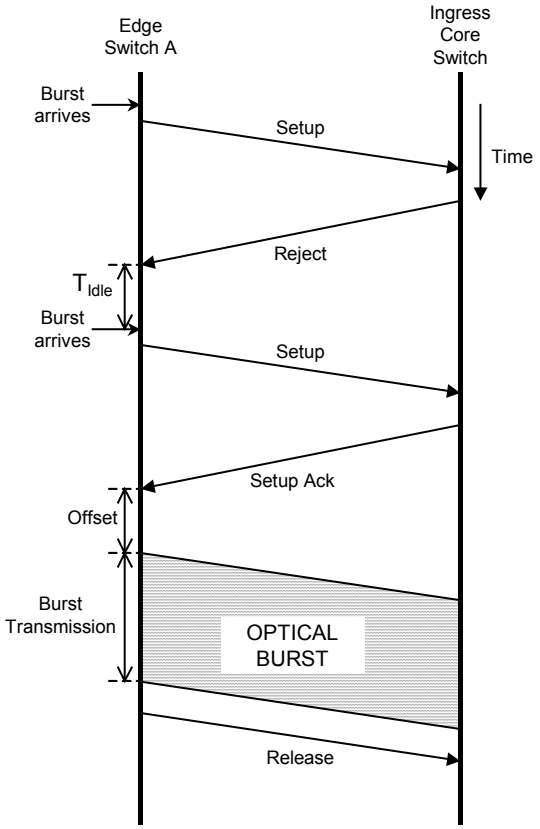


Fig. 2.9. Schematic representation of signaling messages exchanged between an ingress edge node and its ingress core switch using JumpStart protocol.

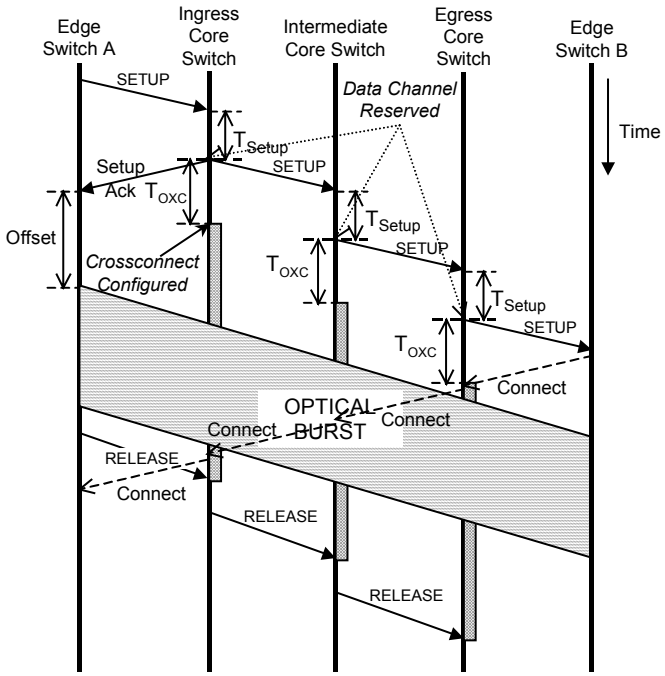


Fig. 2.10. Schematic representation of JumpStart signaling messages for a successful burst transmission.

JumpStart uses control messages to define OXC behavior and thus implement its particular characteristics. This protocol is designed to be implemented in hardware and does not use a scheduler [75, 84].

2.4.2.3 Horizon

The Horizon resource reservation protocol was proposed by Turner in a technical report presented in 1997 [54] on the scope of the Terabit Burst Switching project [185] and published in 1999 [1].

This resource reservation protocol introduces the concept of "Time Horizon" for a given channel and it is called *Horizon* because every data channel has a time horizon during which it is reserved. Time horizon is defined as "the earliest time to which there is no prevision to use the channel (wavelength)" [1]. In other words, time horizon is the earliest time (*deadline*) in which a channel is ready to accept new bursts. In this protocol, the setup message contains both the offset time and the burst length, allowing the computation of the time horizon. This concept is used in other protocols with one-way resource reservation schemes such as JET and JIT$^+$ that are considered in this section.

In Horizon, an output channel is reserved for a burst only if the arrival of the burst happens after the time horizon for that channel; if upon the arrival of the setup message, the time horizon for that channel is later than the predicted arrival time of the first bit of the burst, then, the setup message is rejected and the corresponding burst is lost.

Both Horizon and JET protocols are delayed reservation schemes. However, unlike JET, Horizon does not perform void filling. Figure 2.11 illustrates the operation of Horizon protocol exemplifying delayed reservation schemes. As it may be seen in this figure, after processing the setup message (T_{Setup}), the data channel is still idle (in void state). Little time before the arrival of the correspondent data burst, the OXC is configured (T_{OXC}) to switch it and, when the time horizon is reached, the data channel is released (estimated release).

When a burst is scheduled for a given data channel, the horizon for that channel is updated to the time the burst leaves the node plus the T_{OXC}. Hence, in

Horizon, a burst can only be scheduled if and only if the first bit of the burst arrives after all the bursts scheduled for that data channel have been sent [80].

Figures 2.11 and 2.12 illustrate the operation of Horizon resource reservation protocol showing two successive bursts (*Burst A* and *Burst B*) and the intention of receiving a third (in Figure 2.12).

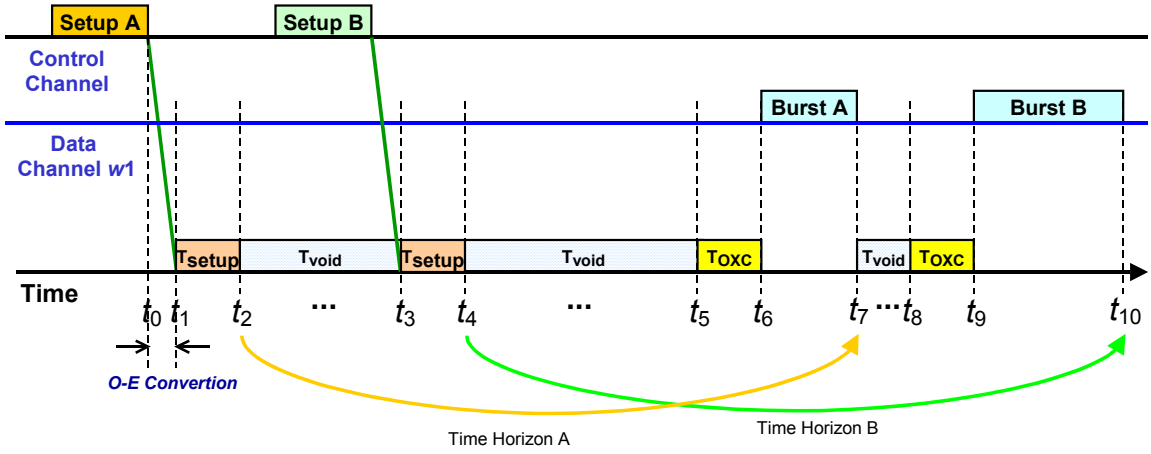


Fig. 2.11. Operation of Horizon resource reservation protocol.

As it may be seen in Figure 2.11, at time t_2 the OXC processed the setup message (*Setup A*) of data *Burst A* and it knows that the time horizon (*Time Horizon A*) for that burst is t_7. The OXC has every data (*Time Horizon A* [t_7], burst duration - *Burst A* [t_7- t_6], and T_{OXC} [t_6- t_5]) to calculate the instant to start the configuration of its switch fabric (that one call $T_{StartOXC}$ - at time t_5). Then,

$$T_{StartOXC}(Y) = Time\_Horizon(Y) - Burst(Y) - T_{OXC} \qquad (2.4)$$

$T_{StartOXC}(Y)$ is the time at which OXC starts the configuration of its switch fabric for switching the *data burst Y*; *Y* identifies the data burst that will be switched. This time is important in one-way delayed reservation protocols in order to schedule the resources reservation.

Between t_2 and t_7, the data channel w_1 is *Void* but OXC only can schedule a new burst if the *time horizon* of that burst is such that the time to start the configuration of this data channel ($T_{StartOXC}$) is greater than t_7. Figure 2.12 shows that the second burst (*Burst B*) fulfils these conditions. That is, the setup message of *Burst B* arrives at time t_3 ($t_2<t_3<t_7$) and its time horizon (t_{10}) is such that the

$T_{StartOXC}$ for *Burst B* (t_8) is great than t_7. Therefore, the setup message (*Setup B*) can be accepted and its respective burst (*Burst B*) can be switched.

Figure 2.12 shows a third burst (*Burst C*) that cannot be switched because the time horizon of that burst (*Time Horizon C* – between t_9 and t_{10}) is smaller than *Time Horizon B*. Thus, *Setup C* is rejected and the corresponding data burst (*Burst C*) is lost.

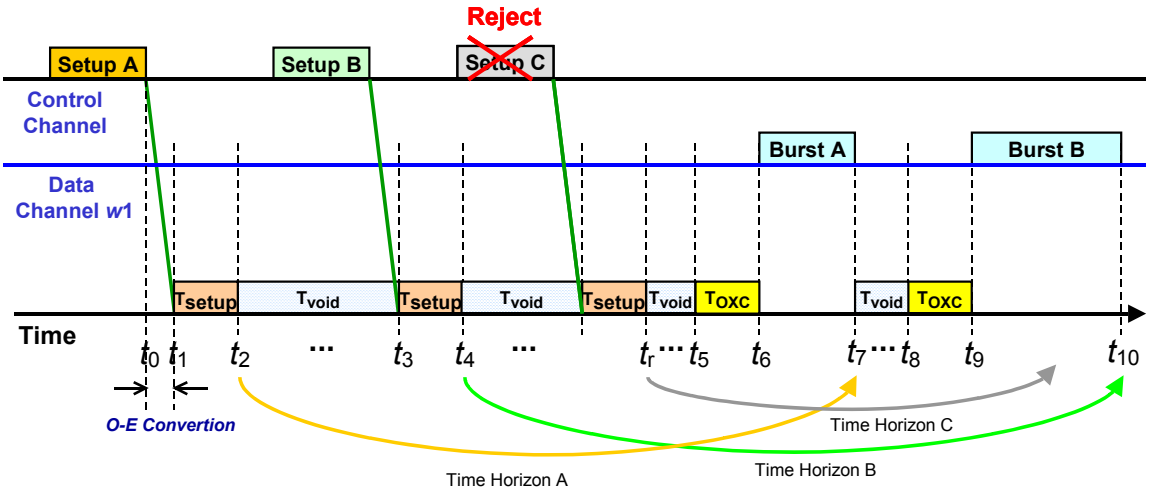


Fig. 2.12. Operation of Horizon resource reservation protocol (rejecting a burst).

Both Horizon and JIT use first-come, first-served (FCFS) service (see Figures 2.11 and 2.12). In terms of burst scheduling complexity, Horizon presents more complexity than JIT. As mentioned in [1, 193], each OBS node needs to maintain the end of the latest reservation (time horizon) for each data channel and a list of time horizons in increasing order.

2.4.2.4 Just-Enough-Time (JET)

Just-Enough-Time (JET) resource reservation protocol was proposed by Qiao in 1997 [55] and published by Qiao and Yoo in 1998 [81] and 1999 [57]. Under JET, an output channel is reserved for a burst only if the arrival of the burst (1) happens after the time horizon defined for that channel, **or** (2) coincides with an idle state (*Void*) for that channel, and the end of the burst (plus the T_{OXC}) is sooner than the end of the idle interval; if, when the *Setup message* arrives, it is determined that

none of these conditions are met for any channel, then the setup message is rejected and the corresponding burst is lost.

JET is the best known resource reservation protocol having a delayed reservation scheme with void filling (idle state), which uses information (from the setup message) to predict the start and the end of the burst. A *Setup message* contains the following information: source and destination nodes, burst length, and offset time. The offset time is updated (reduced) at each intermediate node to compensate for the actual time to process the setup message (T_{Setup}) and the time to configure the OXC (T_{OXC}) [63].

This protocol allows the implementation of a non-FCFS service. If the T_{Offset} of two bursts are similar, then the protocol tends to follow an FCFS order, following the way that Horizon protocol uses to schedule bursts; but if OXC receives a message with a comparatively big T_{Offset}, and afterwards a message with a small T_{Offset} is received, then there is the possibility that OXC serves the burst that releases the channel first (performing void filling). This situation is illustrated in Figure 2.13.

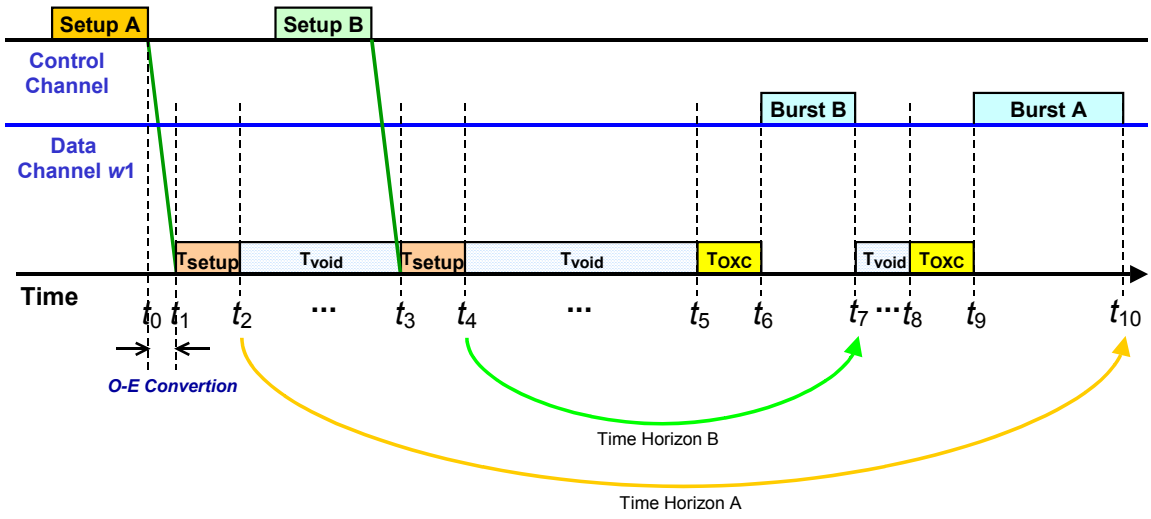


Fig. 2.13. Operation of JET resource reservation protocol.

As it may be seen in Figure 2.13, there are two bursts (*Burst A* and *Burst B*) that are transmitted on the same data channel w_1. T_{Offset} of *Burst A* is long, while T_{Offset} of *Burst B* is short, and the time horizon of *Burst B* (*Time Horizon B* - t_7) is smaller than $T_{StartOXC}$ (t_8) of *Burst A*. Then the OXC can switch the *Burst B* before switching *Burst A*. To do this, after processing the setup message of *Burst B* (*Setup B*) at time t_4, the switch detects that *Burst B* will arrive before the time of starting the

configuration of this data channel ($T_{StartOXC}$) for *Burst A*, and runs a void filling algorithm [95, 193] to verify if it can accept the new burst. In the same situation under Horizon protocol, *Setup B* would be rejected and the corresponding data burst would be lost (*Burst B*).

After this observation, it is expected that JET may perform better than Horizon in terms of burst loss probability. However, an empty space analysis algorithm demands more machine effort in JET than in Horizon [80]. Chapters 4 and 5 will focus on the performance of these resource reservation protocols.

Figures 2.13 and 2.14 show that setup message of *Burst B* (*Setup B*) arrived after setup message of *Burst A* (*Setup A*), and *Burst B* is switched before *Burst A*. This operation results and illustrates that JET uses a non-FCFS service. JET based systems can include some common scheduling algorithms, such as first-fit unscheduled channel (FFUC) [95], latest available unscheduled channel (LAUC) [194], also mentioned as Horizon scheduling [1], and latest available unscheduled channel with void filling (LAUC-VF) [95]. These algorithms are described on Section 2.4.4.

Figure 2.14 illustrates an example of a burst (*Burst C*) that arrives after a second successful burst (*Burst B*). The time horizon of *Burst C* is between t_7 and t_8. Although the data channel w_1 is free between t_7 and t_8, the switch cannot perform this burst because after time t_8 - $T_{StartOXC}$ (A) – since it needs to switch *Burst A*.

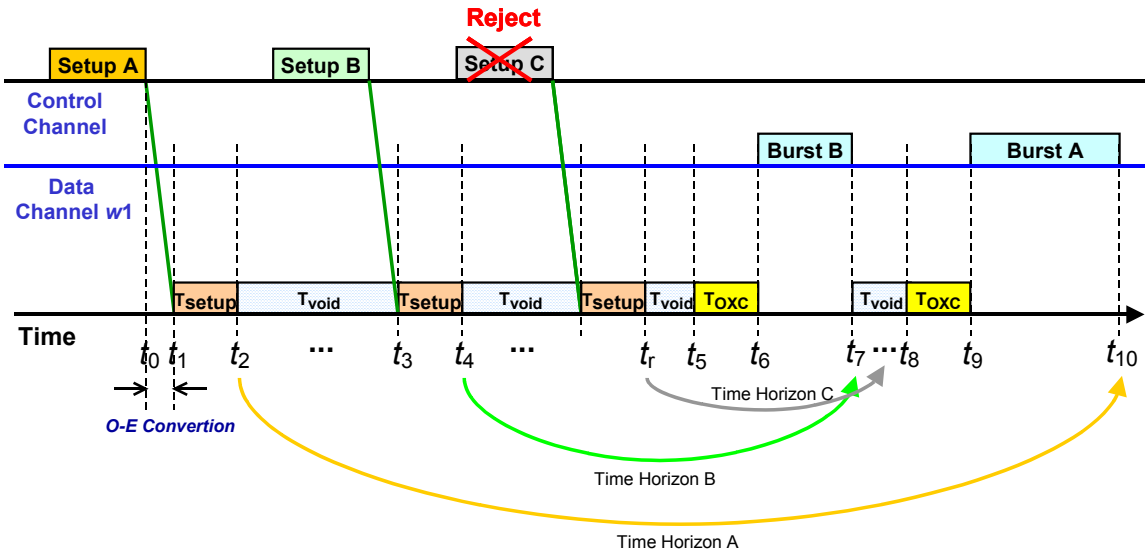


Fig. 2.14. Operation of JET resource reservation protocol (rejecting a burst).

Authors of JET made analytical and simulation studies which confirmed the good effects of delayed reservation on burst loss probability in an OBS network. This resource reservation protocol has been one of the most studied and presented in the literature [4, 6, 47, 58, 60, 63, 75, 80, 133].

A first improvement of JET was proposed in [57] to support quality of service (QoS). Specially, there are two classes of traffic: real-time and non-real-time. Bursts classified as belonging to the real-time class are allocated with higher priority than bursts which belong to the non-real-time class by using an additional delay (extra offset time) between transmission of a control packet and the transmission of the correspondent data burst. The consequence of this additional delay is to reduce burst loss probability of real-time bursts at the optical burst switch.

Yoo, Qiao, and Dixit presented in [195] a new improvement for JET using a priority scheme to support multi-class traffic. In particular, a data burst with higher class is assigned with an additional offset.

2.4.2.5 JIT$^+$

The most recent resource reservation protocol is JIT$^+$ proposed by Teng and Rouskas in 2004 [115] and published in [80]. JIT$^+$ was defined as an improvement of JIT and it combines JIT simplicity with the utilization of the time horizon used by delayed reservation protocols, such as Horizon or JET.

JIT$^+$ is a modified version of JIT protocol, which adds limited burst scheduling (for a maximum of two bursts per channel). Under JIT$^+$, an output channel is reserved for a burst only if (*i*) the arrival time of the burst is later than the time horizon of that data channel and (*ii*) the data channel has at most one other reservation.

Teng and Rouskas [80] consider that it maintains all the advantages of JIT in terms of simplicity of hardware implementation.

Figure 2.15 shows the operation of JIT$^+$ protocol, considering three successive data bursts (*Burst A*, *Burst B*, and *Burst C*) that intend to be switched using data channel w_1. Like in JIT protocol, after processing the first setup message (*Setup A*) at time t_2, the switch configures immediately its switch fabric to switch *Burst A*. The burst is accepted, the time horizon is updated to t_7, and at time t_3 the data channel

w_1 is already configured. At time t_4 a setup message arrives for *Burst B*, as in Horizon, the burst is accepted and the time horizon is updated to t_{11}. Using JIT protocol, the setup message of *Burst B* (*Setup B*) would be rejected.

After time t_5, the data channel w_1 has already one outstanding reservation (for *Burst B*). Then, when the setup message of *Burst C* arrives, at time t_6, this setup message (*Setup C*) is rejected and the correspondent data burst is lost. As Figure 2.15 illustrates, under Horizon or JET, *Setup C* would be accepted and the correspondent data burst (*Burst C*) switched because its time horizon (*Time Horizon C*) is greater than t_{11} (the *Time Horizon B*).

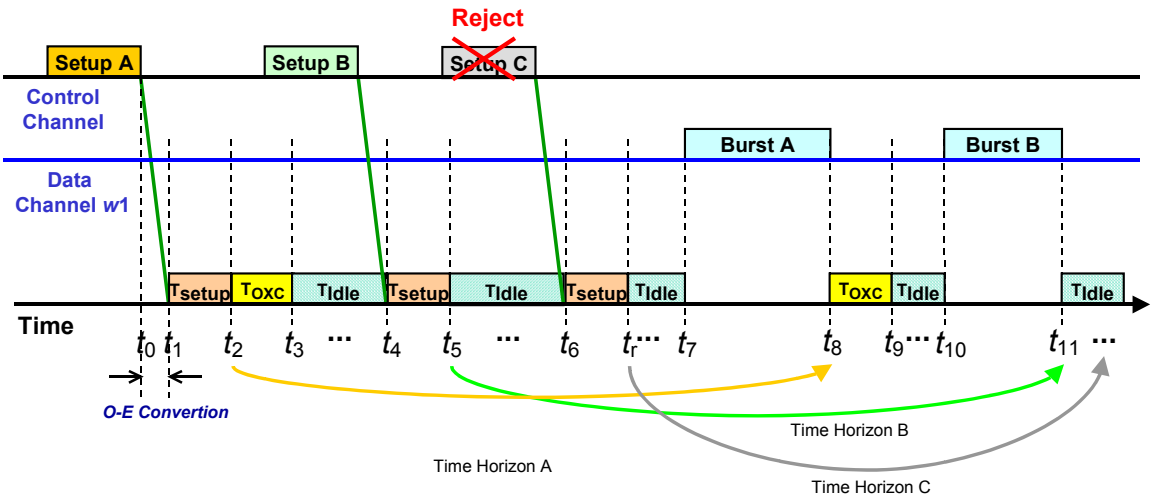


Fig. 2.15. Operation of JIT$^+$ resource reservation protocol (rejecting a burst).

JIT$^+$ protocol is an example of a one-way resource reservation protocol with immediate resources reservation (Figure 2.7). Hence, after switching *Burst A* (between times t_7 and t_8), the OXC is immediately configured (time t_9) to switch *Burst B* (between times t_{10} and t_{11}). Next, after time t_8, it is possible to schedule a new burst.

JIT$^+$, such as Horizon, does not perform empty (void) filling (performed by JET), and tries to improve the performance of JIT doing at the most only one reserve for each data channel (in the case the channel is occupied). Both JIT$^+$ and JIT use first-come, first-served (FCFS) service.

In terms of burst scheduling, JIT$^+$ assigns the first data channel that can transmit a burst (FCFS), such as JIT.

2.4.3 A New Resource Reservation Protocol: Enhanced JIT (E-JIT)

In this sub-section, a new resource reservation protocol for OBS networks, called *Enhanced Just-in-Time* (E-JIT) is proposed. This proposal is based on the relative performance assessment of JIT, JumpStart, JIT$^+$, JET, and Horizon resource reservation protocols (mentioned in Chapters 4 and 5), and the above discussion regarding the relative complexity of them (JIT is the simplest to implement). E-JIT intends to improve and optimize the traditional JIT protocol, keeping all the advantages of its simplicity in terms of implementation. E-JIT aims to improve data channel scheduling, reducing the period of time in which the data channel remains in "reserved" status, optimizing channel utilization and potentially reducing burst loss probability.

E-JIT assumes an out-of-band signaling and the signaling channel is best-effort link by link. It implements the same resource reservation protocol functions that usually are used in one-way OBS reservation protocols such as JumpStart [75, 82, 83]. These functions are described in Table 2.1.

Session Declaration	Announces a new connection to the network
Path Setup	Configures resources needed to establish an all optical path from source to destination
Data Transmission	Informs intermediate nodes of burst arrival time and length
State Maintenance	Refreshes the state information to maintain the connection
Path Release	Releases resources used to transmit the burst

Table 2.1. Signaling protocol functions.

E-JIT supports both short and long bursts. A setup message has the responsibility of Session Declaration, Path Setup and Data Transmission informing intermediate nodes for a burst arrival. Especially for long bursts, to maintain the connection (State Maintenance), the source edge node may also send KEEPALIVE

messages to the intermediate nodes along the path to prevent the switch state from timing out.

E-JIT protocol uses estimate (or implicit) release to set free the switch fabric resources. Setup message also carries the information of burst length and burst offset length. When a burst leaves an OXC, this automatically frees its resources using an estimate release, like JET. Thus, this OXC can immediately accept and switch a new burst. When a burst arrives to its destination edge node, all OXCs along the path from source to destination were successively ready (in a "free" state) to switch a new burst. Otherwise, when any network element detects a "setup" failure, it sends a "release message" to all network elements along the path to the source node. Here, it is assumed that, as mentioned above, the status of data channel w_1 is updated to "free" when the last bit of the burst arrives at the switch and it is intended to present the operation of resources allocation using this protocol. Therefore, in this context, the representation of the implicit "release message" can be neglected. An explicit release message is only sent if any detection of "setup" failure occurs during an end-to-end transmission along the path. Due to lack of space and multiple combinations of possible failures, these situations are not shown in the figures. To calculate the initial offset time, the equation (2.1) used by JIT is used. The message flows for E-JIT is illustrated in Figure 2.16.

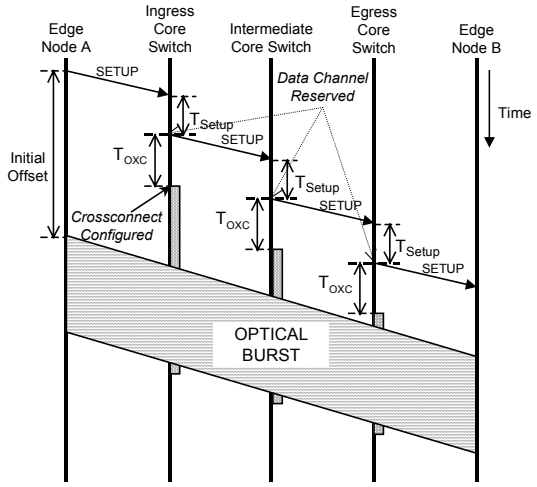


Fig. 2.16. Schematic representation of E-JIT signaling messages for a successful burst transmission.

In terms of system architecture, the description provided in [82, 83] for JIT architecture is followed. Figure 2.17 plots system architecture showing the

relationship between different protocol entities. In the Upper Layer, DownStream indicates that the traffic flows from source edge node to destination edge node, and UpStream designates that the traffic flows from destination edge node to source edge node. At a source edge node, an Upper Layer source edge starts a transition by sending an Open message through the ChanUpper. At the JIT layer, as soon as it receives an Open message from the Upper Layer, the source edge node generates a Setup message using the ChanNSDown. ChanNSUp is used to send messages from ingress core switch to the source edge node. As may be seen in the figure, ChanNSDown and ChanNSUp determine the direction of the connection. ChanSSDown and ChanSSUp represent channels between ingress core switch and intermediate core switch, one for each direction, while ChanXSDown and ChanXSUp symbolize channels between two intermediate nodes.

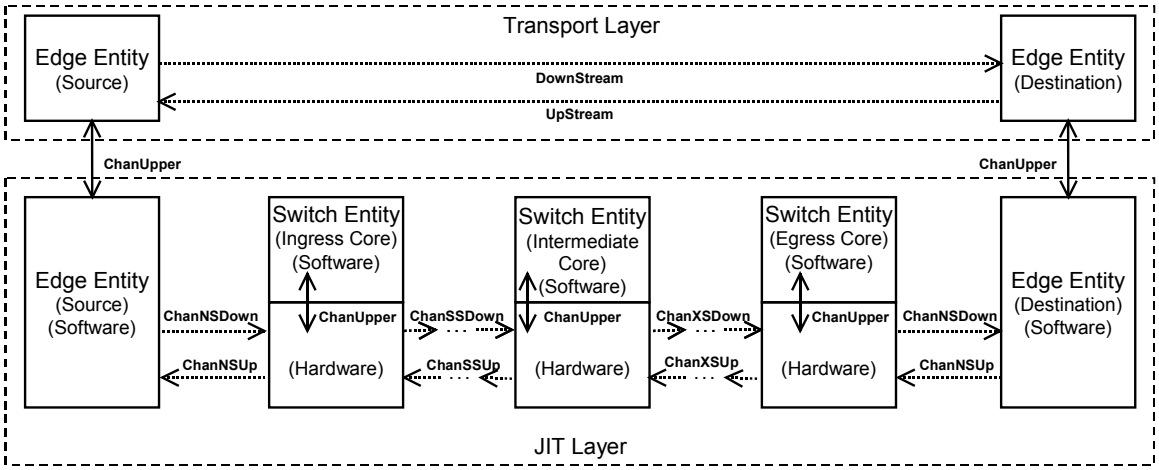


Fig. 2.17. Schematic representation of protocol stack architecture [82].

The protocol operation and its formal specification are provided in the next two Sub-sections.

2.4.3.1 Protocol Operation

Under E-JIT, an output data channel is reserved for a burst immediately after the arrival of the corresponding setup message, if (*i*) this data channel is free or (*ii*) if it is reserved, the *end time* of the last switched burst is smaller than the actual time to process the setup message ($\leq T_{Setup}$). If a channel cannot be reserved

immediately, then the setup message is rejected and the corresponding burst is lost. The operation of E-JIT resource reservation protocol is illustrated in Figure 2.18.

Each data channel may have two states: “free” or “reserved”. If the wanted data channel is “free”, the setup message and the corresponding data burst are accepted. If the wanted data channel is “reserved” and the *end time* of the last switched burst is smaller than the *end processing time* of the incoming setup message, the new setup message and the corresponding data burst are accepted. Otherwise, the setup message is rejected and the correspondent data burst is lost.

Both JIT and E-JIT are resource reservation protocols with immediate resources reservation. This proposal pretends to optimize JIT protocol, reducing free time periods of data channels (until zero, if it is possible – as it may be seen in Figure 2.18) and increasing the data channel utilization. Consequently, it is possible to reduce data burst loss probability and to increase OBS network performance.

Figures 2.18, 2.19, and 2.20 show, in detail, the operation of E-JIT resource reservation protocol. In these figures, the parameters mentioned in Section 2.4.2, are used:

- T_{Setup} represents the amount of time that is needed to process the setup message in an OBS node. It is assumed that this time is equal in every node in the network, since all of them are configured under the same protocol.
- T_{OXC} is the time OXC needs to configure its switch fabric to establish a connection between an input port and an output port. This may also be viewed as the time OXC takes, after interpreting the command in the control message, to position micro-mirrors correctly (MEMS switch [163, 187]) in the matrix and to be used to switch a burst.
- T_{idle} is the idle time in an output channel in an OBS node. During this period of time, OXC is already configured and it continues waiting to switch a burst.

In the figures below (Figures 2.18-2.20), a single data channel of an OBS node is considered, to present the operation of resource reservation protocol. A temporal line is used to illustrate the sequence of events. On this time line, this sequence of events leading to the reservation of a data channel to switch a burst can be best observed. The time line does not show each time in terms of the same scale, due to space limitations (e.g. T_{OXC} is larger than T_{Setup}).

At instant t_0, it is considered that OXC receives the setup message (*Setup A*) and, between t_0 and t_1 converts it from an optical to an electronic signal to be

processed. Instant t_1 is the minimum time that one may consider for the data channel w_1 of OXC to be occupied.

Figure 2.18 shows two successive bursts (*Burst A* and *Burst B*) to be switched. For *Burst A*, it operates like JIT. At instant t_5 arrives a new setup message (*Setup B*) and at instant t_7 it is assumed that OXC processed it. At this time it is possible to accept and switch this burst because $t_7 > t_6$ (t_6 is the end of the *Burst A*). Under JIT *Burst B* would be rejected.

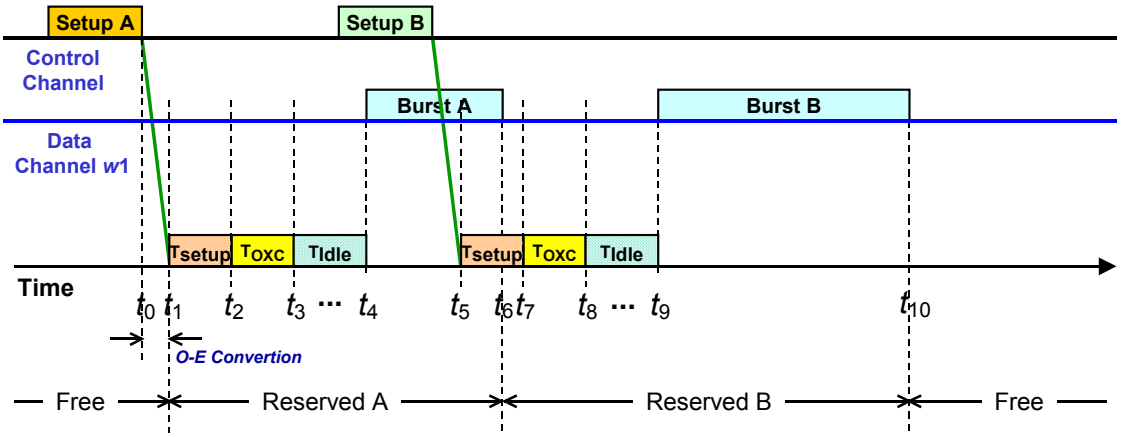


Fig. 2.18. Operation of E-JIT resource reservation protocol.

After t_6, data channel w_1 continues reserved (*Reserved B*) because it will switch *Burst B*. Immediately, after processing the setup message (*Setup B*) at time t_7, between instants t_7 and t_8 OXC is configured to switch *Burst B* (immediate resources reservation). The burst is switched between t_9 and t_{10}. After t_{10} (end of the *Burst B*) the state of the data channel changes to "free" again waiting for a new burst.

The change of state to "reserved" takes place upon receipt of the setup message, whereas the change to state "free" occurs when the last bit of the burst arrives at the switch at time t_5. The length of state "reserved" is equal to the burst length plus the correspondent remaining offset time (T_{Offset}). The length of a state "free" is equal to the time until the arrival of next setup message. If OXC receives a setup message and the *end processing time* of this setup message is greater than the *end switch time* of the last burst, the state remains "reserved" for the new burst (see Figure 2.18).

Figure 2.19 addresses the same sequence of two incoming bursts (*Burst A* and *Burst B*) mentioned above and the intention to receive a third (*Burst C*). In this case,

OXC cannot accept the setup message of the third burst (*Setup C*) because the *end processing time* (t_{10}) of the new setup message (*Setup C*) is less than the *end time* of *Burst B* (t_{11}).

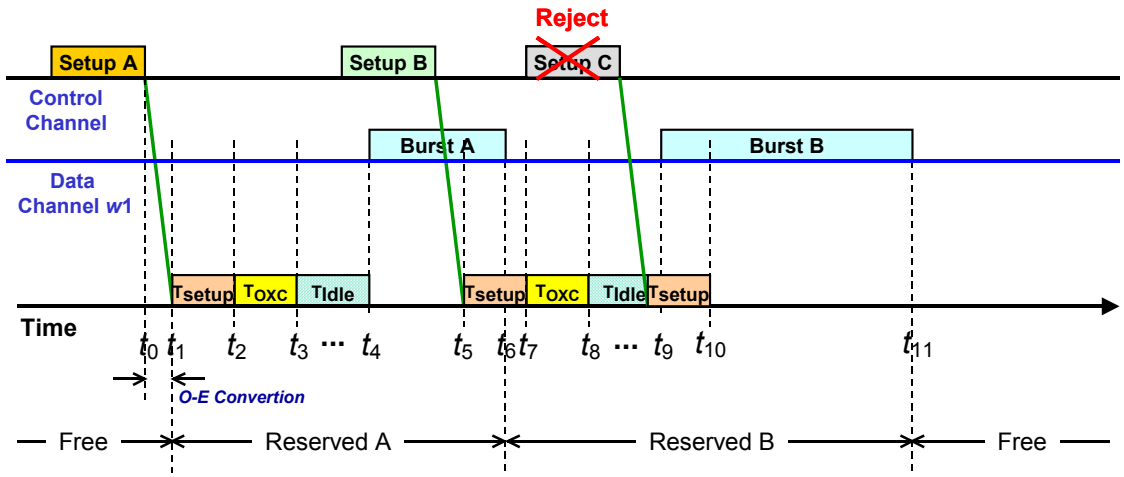


Fig. 2.19. Operation of E-JIT resource reservation protocol (rejecting a burst).

Figure 2.20 shows the arrival of a new burst and the correspondent setup message (*Setup D*), in the sequence of Figures 2.18 and 2.19. When a new setup message arrives to OXC, at time t_{9b}, this setup message is accepted because its *end setup time* (t_{10}) is greater than the *end switch time* (t_{9c}) of the last switched burst (*Burst B*). Such as the case of switching *Burst B* (at time t_6), the state remains "reserved" changing for *Burst D* at time t_{9c}.

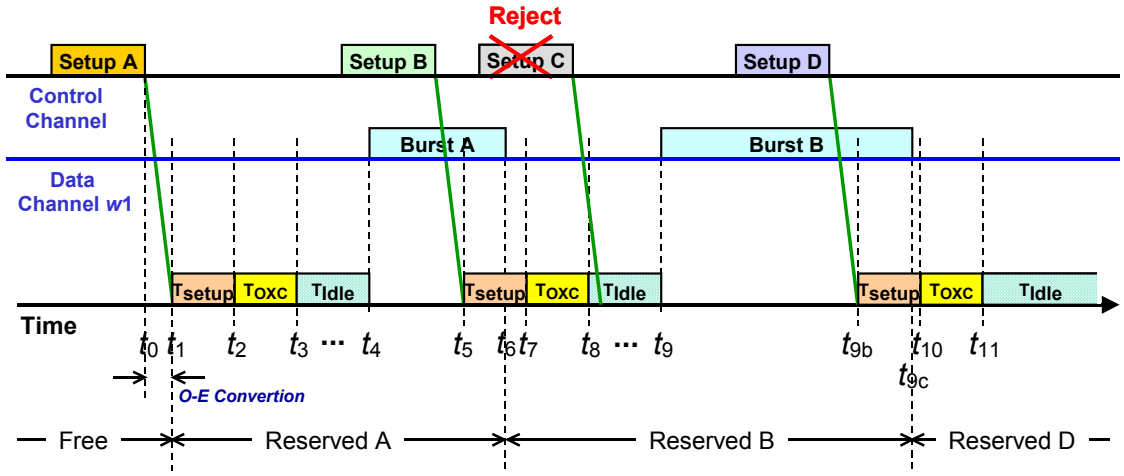


Fig. 2.20. Operation of E-JIT resource reservation protocol (accepting a new burst).

Both JIT and E-JIT resource reservation protocols use FCFS service. Bursts are switched in the order in which their corresponding *setup messages* arrive at the

node. Similar to JIT$^+$, E-JIT schedules the first available data channel to accommodate a burst.

2.4.3.2 Formal Specification

The formal specification of E-JIT follows the extended finite state machine (EFSM) proposed in [82, 83] for formal description of JumpStart JIT signaling protocol. This approach follows this proposal because both E-JIT and JumpSart are JIT based protocols and E-JIT intends to improve the performance of JIT protocol, as mentioned above. EFSM model is very useful because it is full expressive and a good means to describe a communication protocol.

Using this model, each EFSM can be formally represented by an eight-tuple (Σ,*S*,*s*,*V*,*E*,*T*,*A*,δ), where [82, 83]:

Σ – set of messages that can be sent or received;

S – set of states;

s – initial state;

V – set of variables;

E – set of predicates that operate on variables;

T – set of timers;

A – set of actions that operate on variables;

δ - set of state transition functions, where each state transition is represented as follows: $\Sigma^*E(V)^*T \rightarrow \Sigma^*A(V)^*S$.

In terms of transitions, there are two types: spontaneous and "when" transitions. A spontaneous transition does not have an input event on its condition part. A "when" transition has an input event satisfying the *T* condition.

Each transition *T* (an outgoing transition) is represented by $S_1 \rightarrow S_2$, where S_1 is the previous state and S_2 is the following state. A transition is executed when an input event is available and a condition is true. Each transition has two parts: a condition part and action part. The condition part has an input event and a Boolean expression that expresses a condition (expressed in the nominator). The action part may be an output event or a statement operating on variables (expressed in the denominator). In each transition, ?chan. *m* represents an input message from given

channel carrying the message *m*, and !chan. *m* denotes an output message to the indicated channel carrying the message *m*. Settimer(*T*,*C*) is an action that sets the timer *T* to a value expressed by *C*.

An example of a spontaneous transition is: $T1: \frac{}{\begin{array}{l}\text{var}\,1 := FALSE \\ Settimer(T1, Const)\end{array}}$

An example of a "when" transition is: $T2: \frac{?chanA1.Continue}{\begin{array}{l}Settimer(T1, Const) \\ !chanD1.Stop\end{array}}$

Following the EFSM model, the state diagrams for source edge node, destination edge node, ingress core node and intermediate core node are defined. To improve reading, each arc of the state diagrams represents a set of transitions, and these transitions are shown in different figures. The state transitions use the different channels that were shown in Figure 2.17.

The state machine for source edge node is the first to define. Thus, the set of messages is:

Σ={Open,Setup,Failure,Timeout,Connection_Failure,Close,Release,
Clear_to_Send,Transmission_Complete,Keepalive}.

Open is created by Transport Layer to notify JIT Layer incoming of a burst. Source Edge node at JIT Layer generates the Setup message for resources reservation. Failure can be generated by any node to notify an error. Timeout is used for each timer defined within the Timeout message. Connection_Failure is used to notify upper layers that an error occurred during the connection phase. Close is used to release the connection (at the Software Layer). Release is the message sent to set free switch resources along the path (at the Hardware Layer). Source Edge node JIT Layer uses Clear_to_Send message to notify Transport Layer that the setup phase is complete and the corresponding burst can be sent. Transmission_Complete notifies Upper Layer that the transmission of a burst has been completed successfully. Keepalive is used for long bursts to maintain the connection until burst ends.

The set of states *S* is:

S={IDLE,SETUP_PROCEEDING,DATA_TRANSMISSION}.

The state machine waits at IDLE state for an Open message notifying incoming of a burst. After that, the machine goes to SETUP_PROCEEDING state and stays there for the duration given in Burst_Delay. The machine moves to DATA_TRANSMISSION when Setup_Timer times out indicating the beginning of the data burst. When

Burst_Timer times out, the connection is closed, the machine returns to IDLE state, waiting a new request.

IDLE state is the initial state *s* and its set of variables is:

V={Burst_Delay,Burst_Time,KA_Time}.

Burst_Delay is the required delay at source node before sending the burst (initial offset time). Burst_Time is the duration of the burst (burst length or burst size). KA_Time is the Keepalive timer set up to send keepalive messages.

The set of timers is:

T={Setup_Timer,Burst_Timer,KA_Timer}.

Setup_Timer is the Setup message processing time (T_{Setup}). Burst_Timer and KA_Timer are the timers of Burst_Time and KA_Time, respectively. ChanT1 is the Timer channel.

The set of actions that operate on variables is:

A={Settimer,Update}.

Figure 2.21 and Table 2.2 show the state diagram and the state transitions for Source Edge Node, respectively. The state diagram waits in the IDLE state until an Open message arrives from the Upper Layer. Then, the source edge node generates a Setup message with the following two variables: Burst_Delay and Burst_Time. These variables are used to set the Burst_Timer. Burst_Delay is updated by function Update at each hop subtracting the processing time from Burst_Delay.

Once receiving an Open message the state changes to SETUP_PROCEEDING setting the timer Setup_Timer. In the SETUP_PROCEEDING state, it can return to IDLE state if it receives a Close message from Upper Layer (forcing connection tear-down) or a Failure message from Ingress Core Node. Otherwise, it waits until Setup_Timer times out and changes to DATA_TRANSMISSION state. In the DATA_TRANSMISSION state, it can also go to IDLE state if it receives a Close message from Upper Layer (forcing connection tear-down) or a Failure message from Ingress Core Node. If KA_Timer times out it stay in the same state, resetting KA_Timer and sending Keepalive messages to maintain the connection. When Burst_Timer times out, the connection is closed and the state changes to IDLE state, waiting for a new Open message.

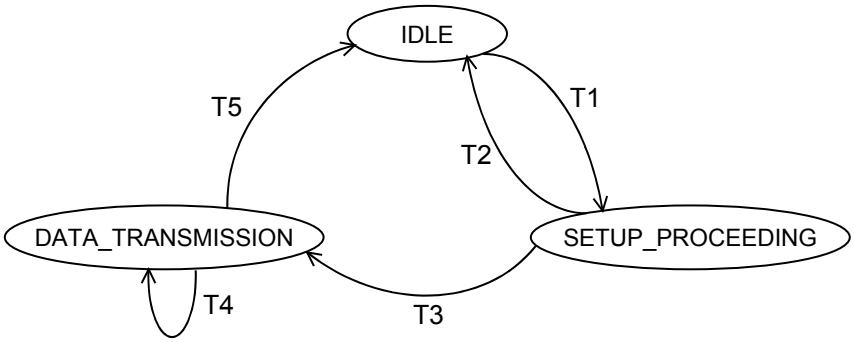


Fig. 2.21. State diagram for source edge node.

State	IDLE
Transition	T1: $\frac{?ChanUpper.Open}{!ChanNSDown.Setup(Burst\_Time, Burst\_Delay)}$ $Settimer(Setup\_Timer, Burst\_Delay)$
State	SETUP_PROCEEDING
Transition	T2: $\frac{?ChanUpper.Close}{!ChanNSDown.Release}$ $\frac{?ChanNSUp.Failure}{!ChanUpper.Connection\_Failure}$
Transition	T3: $\frac{?ChanT1.Timeout(Setup\_Timer)}{If(Burst\_Time = Specified)}$ $\{Settimer(Burst\_Timer, Burst\_Time)\}$ $Settimer(KA\_Timer, KA\_Time)$ $!ChanUpper.Clear\_To\_Send$
State	DATA_TRANSMISSION
Transition	T4: $\frac{?ChanT1.Timeout(KA\_Timer)}{Settimer(KA\_Timer, KA\_Time)}$ $!ChanNSDown.Keepalive$
Transition	T5: $\frac{?ChanUpper.Close}{!ChanNSDown.Release}$ $\frac{?ChanNSUp.Failure}{!ChanUpper.Connection\_Failure}$ $\frac{?ChanT1.Timeout(Burst\_Timer)}{!ChanUpper.Transmission\_Complete}$

Table 2.2. State transitions for source edge node.

The next state machine to be described belongs to the Destination Edge Node. This receives the Setup message and the corresponding data burst, and closes the connection. The set of messages is

Σ={Open,Setup,Failure,Timeout,Setup_Complete,Close,Release,
Transmission_Complete,Keepalive}.

A new message in this state diagram is Setup_Complete. This message notifies the Upper Layer that the transmission of a burst has been completed successfully.

The set of states *S* is:

S={IDLE,SETUP_PROCEEDING,DATA_TRANSMISSION}.

Initial state *s* is the IDLE and its set of variables is

V={Burst_Delay,Burst_Time,KA_Time}.

The set of timers is:

T={Burst_Timer,KA_Timer}.

The set of actions that operate on variables is:

A={Settimer,Update}.

Figure 2.22 and Table 2.3 show the state diagram and transitions for this Destination Edge node, respectively. The state diagram waits in the IDLE state until a Setup message arrives. After receiving the Setup message, it changes to the SETUP_PROCEEDING state and the JIT Layer sends an Open message to the Upper Layer. If the Upper Layer answers with a Close message (forcing connection tear-down), then the Destination generates a Failure message to the Source node. Otherwise, it updates the Burst_Delay, sets Burst_Timer and KA_Timer, and changes to DATA_TRANSMISSION state. In the DATA_TRANSMISSION state, it can receive Keepalive messages to maintain the connection (and KA_Timer is reset), or it can go to the IDLE state. This may occur if Burst_Timer or KA_Timer times out, or if Edge node receives a Release message from the last Core node, or if it receives a Close message from the Upper Layer. In the case that it receives a Close message, the protocol generates a Failure message indicating that the Destination Edge node forces the connection teardown.

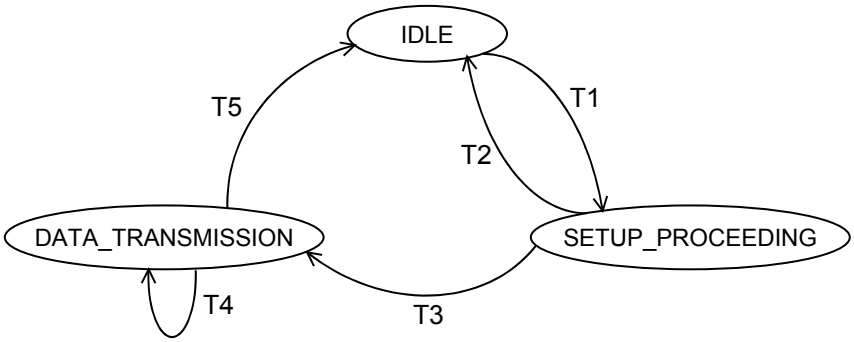


Fig. 2.22. State diagram for destination edge node.

State	IDLE
Transition	T1: $\frac{?ChanNSDown.Setup(Burst\_Time, Burst\_Delay)}{!ChanUpper.Open}$
State	SETUP_PROCEEDING
Transition	T2: $\frac{?ChanUpper.Close}{!ChanNSUp.Failure}$
Transition	T3: $\frac{?ChanUpper.Setup\_Complete}{If(Burst\_Time = Specified) \{Settimer(Burst\_Timer, Burst\_Time + Burst\_Delay)\} \; Settimer(KA\_Timer, KA\_Time)}$
State	DATA_TRANSMISSION
Transition	T4: $\frac{?ChanNSDown.Keepalive}{Settimer(KA\_Timer, KA\_Time)}$
Transition	T5: $\frac{?ChanT1.Timeout(Burst\_Timer)}{!ChanUpper.Transmission\_Complete}$ $\frac{?ChanT1.Timeout(KA\_Timer)}{!ChanUpper.Failure}$ $\frac{?ChanNSDown.Failure}{!ChanUpper.Failure}$ $\frac{?ChanUpper.Close}{!ChanNSUp.Release}$

Table 2.3. State transitions for destination edge node.

Now, the state diagram for a Core switch that receives a setup request from Source Edge node or for another Core switch is described. After receiving a Setup message it configures its switch fabric matrix and chooses the outgoing data channel to transmit the incoming burst. Each Core switch, after processing the Setup message

and making the necessary allocation inside the switch, sends the Setup message to the next switch. The set of messages is:

Σ={Open,Setup,Failure,Timeout,Close,Release,Keepalive}.

A new message in this state diagram is Setup_Complete. This message notifies the Upper Layer that the transmission of a burst has been completed successfully.

The set of states *S* is:

S={IDLE,SETUP_PROCEEDING_ERR_CHECK,DATA_TRANSMISSION}.

A new state, called SETUP_PROCEEDING_ERR_CHECK, has the same functionalities than the previous SETUP_PROCEEDING state, plus running the ErrorCheck function, mentioned below, to verify if there are errors.

Initial state *s* is the IDLE and its set of variables is:

V={Burst_Delay,Burst_Time,KA_Time,ErrorCode}.

The set of timers is:

T={Burst_Timer,KA_Timer}.

The set of actions that operate on variables is:

A={ErrorCheck,Settimer,Update}.

At each intermediate node, after receiving a Setup message, the switch runs some checks to verify if there are errors, such as cyclic redundancy code (CRC), buffer overflow, OXC error, etc. The function ErrorCheck, defined for this state machine returns an error number expressed by the variable ErrorCode. If an error occurs, this variable indicates the type of error and the state machine goes to the IDLE state. If ErrorCode is NULL, the state machine changes to DATA_TRANSMISSION state. This proposed protocol follows the list of possible errors presented in [83], that are listed in Table 2.4.

Error Type	Error Code	Description
no_error	0	The check returns without errors
crc_error	1	CRC error
ime_buf_overflow	2	An ingress core switch message buffer overflow
gme_buf_overflow	3	An intermediate core switch message buffer overflow
sigmess_state	4	A state machine error
sigmess_oxc	5	A optical cross connect error
label_lut	6	A label look-up table error

Table 2.4. Possible error types and codes.

Figure 2.23 and Table 2.5 show the state diagram and transitions for Intermediate (Core) nodes, respectively. Figure 2.17 distinguishes channels between Edge and Core nodes from channels between Core nodes to facilitate reading and comprehension. These names are used in this section to present the state machine of Core nodes, showing an Ingress Core node. However, the behavior of each Core node is similar.

The state diagram waits in the IDLE state until a Setup message arrives. After receiving the Setup message, it changes to the SETUP_PROCEEDING_ERR_CHCK state. Following, the JIT Layer sends an Open message to the Upper Layer and it runs some checks to verify if there are errors, as mentioned above. If errors occur, it sends a Close message to the Upper Layer and a Failure message to the next switch. If the Core switch receives a Release message after receiving a Setup message, it transmits the Release message to the next switch and a Close message to the Upper Layer. If no errors occur, after updating Burst_Delay (by subtracting T_{Setup} from the Burst_Delay variable received in the Setup message) and setting the Burst_Timer, it transmits the Setup message to the next switch. After setting KA_Timer, it moves to DATA_TRANSMISSION state.

In the DATA_TRANSMISSION state, when Burst_Timer times out, the connection may be closed and the state changes to IDLE (with a successful transmission). The Core switch also can receive Keepalive messages to maintain the connection (KA_Timer is reset) and the message is passed to the next switch. If it receives a Release message from the previous node, it can send this message to the next node and a Close message to the Upper Layer, and it goes to IDLE state. Another possibility is to receive a Failure from the next node. In this case, it sends a Close message to the Upper Layer and sends a Failure to the previous node.

At the Core nodes, after receiving a Close message in their Upper Layer, they are ready to receive new bursts, both after transmitting a burst and receiving a Failure message or Error detection.

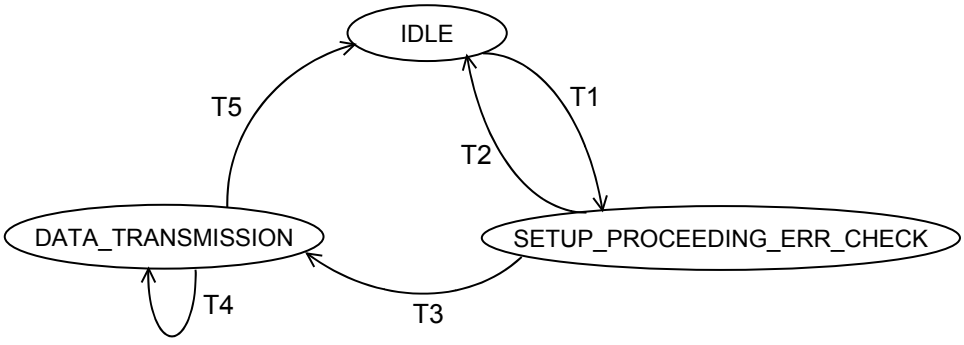


Fig. 2.23. State diagram for core node.

State	IDLE
Transition	T1: $\frac{?ChanNSDown.Setup(Burst\_Time, Burst\_Delay)}{ErrorCheck(ErrorCode)}$ $!ChanUpper.Open$
State	SETUP_PROCEEDING_ERR_CHECK
Transition	T2: $\frac{ErrorCode}{!ChanNSUp.Failure}$ $!ChanUpper.Close$ $\frac{?ChanNSDown.Release}{!ChanSSDown.Release}$ $!Chan.Upper.Close$
Transition	T3: $\frac{NoErrorCode}{!ChanSSDown.Setup(Burst\_Time, Burst\_Delay)}$ $If(Burst\_Time = Specified)$ $\{Settimer(Burst\_Timer, Burst\_Time + Burst\_Delay)\}$ $Settimer(KA\_Timer, KA\_Time)$ $OXC\_Config()$
State	DATA_TRANSMISSION
Transition	T4: $\frac{?ChanNSDown.Keepalive}{!ChanNSDown.Keepalive}$ $Settimer(KA\_Timer, KA\_Time)$
Transition	T5: $\frac{?ChanT1.Timeout(Burst\_Timer)}{!ChanUpper.Close}$ $\frac{?ChanSSUp.Failure}{!ChanNSUp.Failure}$ $!ChanUpper.Close$ $\frac{?ChanNSDown.Failure}{!ChanSSDown.Failure}$ $!ChanUpper.Close$

Table 2.5. State transitions for core node.

2.4.4 Data Channel Scheduling Algorithms

Channel scheduling algorithms determines the manner in which available outgoing data channels are found for an incoming data burst and makes a new reservation for this burst. In sub-section 2.4.2 the data channel reservation were described and operation algorithms for each one-way resource reservation protocol presented. This sub-section is devoted to present several channel scheduling algorithms mentioned in the literature.

Xiong *et al*. [194] presented in 1999 a scheduling algorithm without void filling called Latest Available Unscheduled Channel (LAUC). However, this scheme is very similar to, if not exactly the same as the Horizon protocol presented above in sub-section 2.4.2.3. In 2000, Xiong *et al*. [95] proposed an extension of LAUC (or Horizon), the Latest Available Unscheduled Channel with Void Filling (LAUC-VF). The difference between Horizon and LAUC-VF is that, even if a data channel is scheduled, it is still considered available. This is so because it is possible to switch a short burst into a time gap before the arrival of the next schedule burst, performing void filing (like the JET).

Yang *et al*. [196] proposed a scheduling algorithm, based on an existing LAUC-VF algorithm to support Differentiated Services (DiffServ) and takes advantage of MPLS, which is called generalized LAUC-VF. This algorithm improves QoS performance by prioritizing data bursts, maintaining multiple queues and utilizing limited optical buffers. By simulation, the authors concluded that this algorithm has better QoS performance than the existing LAUC-VF algorithm.

Recently, Xu *et al*. [193] proposed several scheduling algorithms (including Min-SV - *minimum starting void*, Min-EV - *minimum ending void*, Max-SV - *maximum starting void*, Max-EV - *maximum ending void*, Batching FDL, and Best Fit), based on techniques from computational geometry, for scheduling burst in JET architecture (presented in Sub-section 2.4.2.4). Considering the three algorithms with more potential, Min-SV works like LAUC-VF, but it is much faster. Min-EV aims to minimize the gap (void) between the end of a new reservation and an existing reservation, and Best Fit wants to minimize the total length of starting and ending voids created after reservation. The authors used simulation to compare the proposed algorithms with Horizon and LAUC-VF, and concluded that the algorithm Min-SV can schedule bursts as fast as Horizon and it achieves burst loss as low as LAUC-VF.

Dolzer [106, 107] proposed a new combined framework for burst assembly, resources reservation mechanism, and the communication between them, called Assured Horizon. In order to keep core nodes very simple, complexity and intelligence are moved to ingress nodes. At ingress nodes, bursts are classified to forwarding equivalent classes (FECs). Because of simplicity, Horizon signaling protocol was chosen in core nodes. The idea of Assured Horizon for ingress nodes is that if the number of FECs is greater than the number of outgoing wavelengths, collisions have to be avoided. In this case, scheduling algorithms from the electronic domain can be applied because bursts are still waiting in an electronic queue. This protocol uses first-come, first-served (FCFS) scheduling in FECs of the same priority and static priority between FECs of different priorities. Assuming reasonable dimensioning of outgoing optical bandwidth, it can be expected that no burst is lost at ingress node.

2.5 Contention Resolution

Using one-way resource reservation protocols, the ingress node sends out data bursts without receiving reservation acknowledgments or global coordination [63]. Then, at intermediate nodes the problem of possible contention among bursts needs to be solved.

Contention occurs when several bursts contend for the same output channel [93]. In OBS, contention is particularly aggravated by burst size variable and long burst duration [116]. In electronic packet switching, contention is handled through buffering. However, in optical domain it is difficult to implement buffers because there is no equivalent RAM.

When contention occurs, the contending burst can be simply dropped, or dropped and retransmitted [117]. If a drop policy is used, the contending burst is simply dropped and discarded. Then, the switch that discards the burst sends a negative acknowledgement to the source node, which then notifies the source IP router. The source node does not retransmit because, under this policy, retransmission is entirely up to the source IP router or the application that generates the request.

The retransmission policy is similar to the drop policy except for the retransmission that is done at the WDM layer. After receiving the negative acknowledgement, the source switch retransmits the discarded burst, without

notifying the source IP router. However, to improve burst loss probability and OBS network performance there are four main contention resolution techniques: optical buffering using fiber delay lines (FDLs) [44, 46, 47, 57, 118], wavelength conversion [119-122], deflection routing [46, 57, 123-125], and burst segmentation [93, 116, 126-128].

2.5.1 Optical buffering

Fiber delay lines – FDLs [44, 46, 47, 57, 118] (deflection applied in *time domain* [63, 129, 130]) are a solution known to implement optical buffering. An optical delay line can delay a burst for a specific period of time, which is directly related to the length of the delay line. Figure 2.24 presents a schematic representation of FDLs, also referred to as fixed-delay FDL buffer [136]. Each circle represents a unit delay, providing only a limited delay and it cannot perform most useful buffer functions. Depending on the number of delay units that are needed, a burst goes to the correspondent fiber delay line. For instance, delay line *i* delays a burst for *i* time slots; then, the burst will be routed to i^{th} delay line to be delayed *i* units. In contrast with electronic buffers, FDLs only provide a fixed delay and they use first-come, first-served approach, in the sense that bursts leave the FDL in the same order in which they enter. There is an additional hardware cost when an FDL is used in an OBS network.

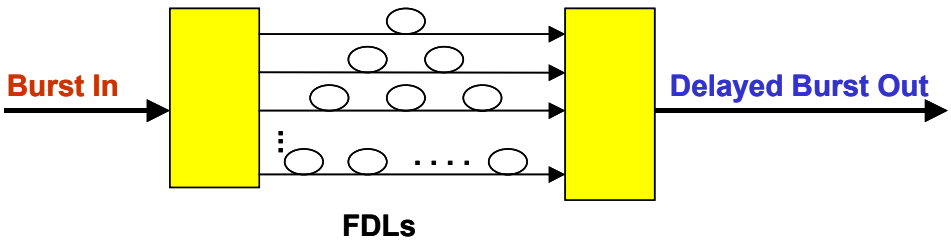


Fig. 2.24. Fiber delay lines (FDLs).

Yoo *et al.* refer in [136] two other types of FDLs called (1) variable-delay FDL buffer and (2) hybrid FDL buffer. The variable-delay FDL buffer has also the same number of delay lines as the fixed-delay, but each delay line can give a variable delay ranging from zero to the maximum of delay provided by the total number of fiber delay lines. This is possible using multiple wavelength-sensitive 2:2 switches.

Thus, the hybrid FDL buffer is a conjunction of fixed-delay and variable-delay FDL buffer. In this FDL buffer, each delay line can provide a variable delay, but it has a different maximum delay ranging from *i* (correspondent to the i^{th} delay line) to the maximum of delay provided by the total number of fiber delay lines.

There are several proposals of switch equipments that include optical buffering, namely, staggering switch [118], switch with FDLs such as contention resolution by delay lines (CORD) [44], and switch with large optical buffers (SLOB) [197].

2.5.2 Wavelength conversion

Wavelength conversion [119-122] (deflection applied in *wavelength domain* [63, 129, 130]) consists in the conversion of the wavelength of an input channel into another wavelength at the output channel. Figure 2.25 illustrates the comparison between networks with: (a) a node without wavelength conversion, and (b) a node with wavelength conversion. In Figure 2.25 in (a), three data channels (λs) were established: i) between node 1 and node 3 on wavelength λ_1; ii) between node 1 and node 2 on wavelength λ_2; and iii) between node 2 and node 3 on wavelength λ_3. Suppose that a data channel between node 1 and node 3 needs to be set up for an incoming burst. Although there is one wavelength available from node 1 to node 2 (λ_3), and one wavelength available from node 2 to node 3 (λ_2), it is impossible to establish a channel for a new incoming burst i.e. that arrives at node 2 through λ_3. This is due to the fact that there is a difference between the two available wavelengths — i.e. from node 1 to node 2 and from node 2 to node 3. As a consequence, the control packet (setup message) of the new data burst (contending burst), requesting this channel, is rejected. Therefore, wavelength selective networks may have higher burst loss than wavelength interchanging networks. The contention can be resolved by the use of wavelength conversion, as shown in Figure 2.25 in (b). In terms of circuit switching, when a relatively small fraction of nodes is equipped with wavelength converters, the network is referred to as one with sparse wavelength conversion [198].

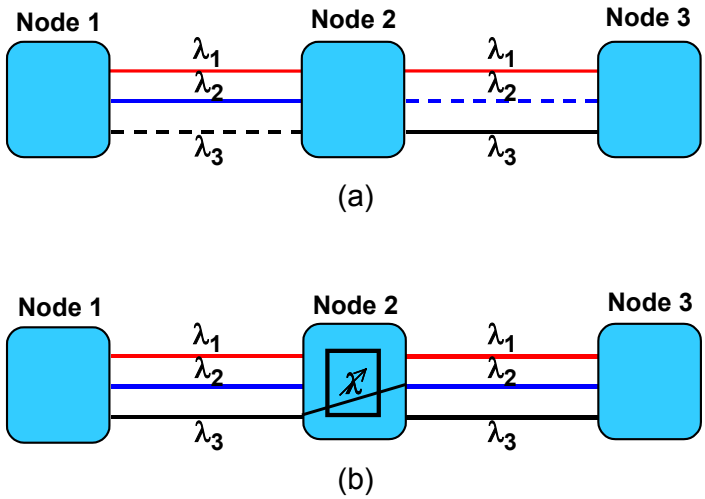


Fig. 2.25. Wavelength conversion in OBS networks; (a) network without wavelength conversion; (b) network with wavelength conversion at node 2.

Gauger [130] proposed a new contention resolution technique based on wavelength conversion, but he did not use full wavelength conversion, taking into account that wavelength converters are technologically complex and expensive. The author analyzed the performance of an OBS node that employs only partial wavelength conversion, in the sense that wavelength converters are only available for a limited number of bursts at a time. He considers the utilization of tunable wavelength converters, shared in a converter pool, and these converters may be assigned to a wavelength channel in case of contention. This study analyzed the combination of a converter pool with FDLs buffers. It concluded that a strategy that prefers FDLs over converters for contention resolution improves performance loss for converter pools with a small number of converters.

2.5.3 Deflecting routing

Deflecting routing was proposed for contention resolution in the context of OBS network by Wang *et al.* [123] in 2000. Under deflecting routing scheme [46, 57, 123-125] (deflection applied in *space domain* [63, 129, 130]), the contending burst is routed to an alternative port, and then the burst follows an alternative route to the destination. Using deflection, the contending burst outgoes through an alternative path; then it takes a higher delay than if it would have gone through shortest path. However, when an alternative link is chosen for deflection, the “shortest path – to

destination – first" is used. If there are no alternative ports available, the contending burst is discarded as in a drop policy [117]. A drawback of deflection is the possibility of looping if the burst outgoes always by the same alternative channels without finding its destination node.

Figure 2.26 shows an example of the effect of deflection routing in contention resolution, in an OBS network with six nodes and eight links. The incoming burst, from source node (node 1) to destination node (node 4), contends at node 3. Following the retransmission policy, the burst needs a two hop distance from source node until node 3 (when contention occurs) and then another three hops distance from source to destination node. Thus, the total hop distance is (2 + 3 = 5) five hops. If deflection is available, (2 + deflection hop count '2' = 4) four hops are enough to resolve the contention and to route the burst to its destination (node 4). In most cases, the total number of hops is smaller than in the case of retransmission. In networks with large number of nodes this effect becomes more obvious.

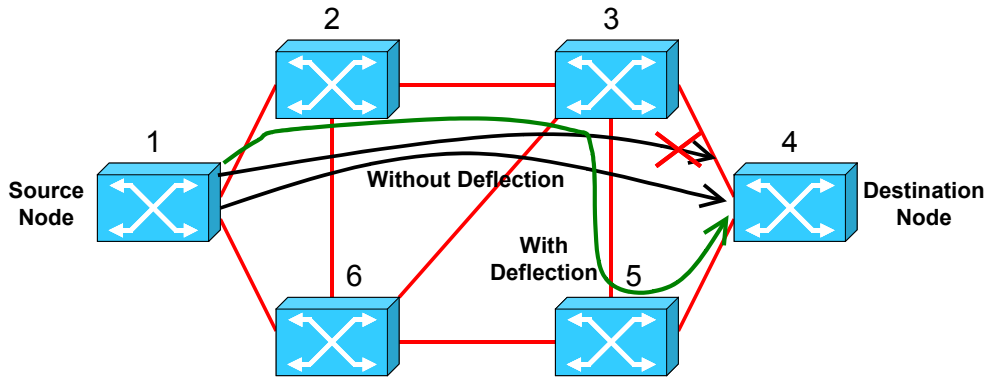


Fig. 2.26. Effect of deflection routing in contention resolution.

A simplification of deflection routing can be obtained with hot-potato routing [119], which presents a special implementation of deflecting routing without the utilization of optical buffers.

Recently, in [199] a study of the impacts brought by deflecting routing in JET-based OBS networks was presented, which concluded that, if reflecting routing is enabled, optical buffers are needed to solve the insufficient offset time problem.

Deflecting routing technique can reduce burst loss and the average delay comparing it with burst retransmission from the source or burst drop policies, mainly, in long distance links [125].

2.5.4 Burst segmentation

Burst segmentation [93, 116, 126-128] was introduced to reduce burst loss probability in OBS networks [116, 126]. This technique follows the principle that it is better to partition the burst into multiple segments and only drops overlapping segments than dropping the entire burst during the contention.

Vokkarane *et al*. [116, 126] proposed a technique based on the concept of burst segmentation, in which each burst consists in a number of basic transport units called segments. A segment header and the respective payload make each segment. The segment header contains the following fields: synchronization bits, error correction information, source and destination information, and the length of the segment (if segments are of variable length). The segment payload carries data, such as IP packets or ATM cells.

When contention occurs, a burst is divided into multiple segments, and only a few segments of the contending bursts will be dropped. Vokkarane *et al*. [116, 126] consider two main approaches for dropping burst segments:

3. *Tail-dropping* - drops the tail of the original burst (Figure 2.27), or
4. *Head-dropping* - drops the head of the contending burst.

The tail-dropping approach is usually chosen because it has the advantage of having a better chance of in-sequence delivery of packets at destination, assuming that the dropped segments (data packets) are retransmitted at a later time. If head-dropping is used, it is likely that the dropped segments will arrive at their destination out of order.

Figure 2.27 illustrates an example of burst segmentation using tail-dropping approach. It illustrates two contending bursts arriving and it represents the selection of the segments to be dropped. Using a tail-dropping policy, tail segments of the original burst that belong to the contention region are dropped. The authors of this proposal consider that, if the switching configuration time is non-negligible, then additional segments may be lost when the switch is configured from one burst (the original burst) to another (the contending burst).

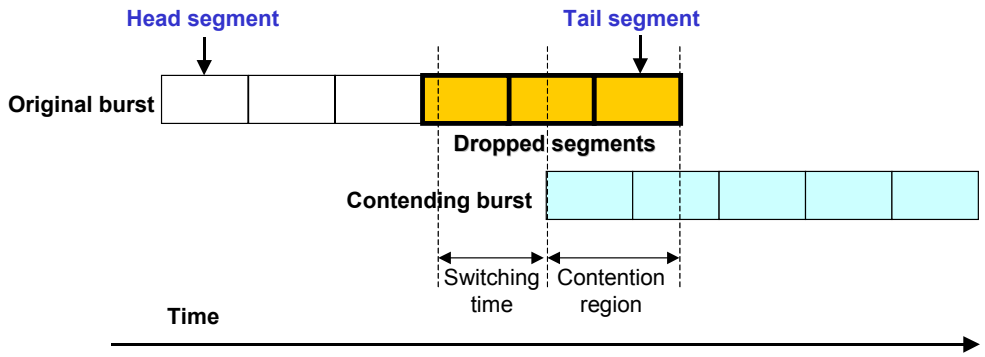


Fig. 2.27. Contention resolution using burst segmentation for two contending bursts.

Vokkarane *et al*. [116, 126] present a modified version of tail-dropping policy to decide which segments of the contending burst are dropped. Under this policy, the tail of the original burst is dropped only if the number of the segments in the tail is less than the total number of the segments of the contending burst. Otherwise, the entire contending burst is dropped.

Maach and Bochmann [131] proposed a new segmented based contention resolution. At the ingress edge node packets are assembled into a segmented bursts made of a list of segments with constant length (one to five IP packets in a segment). The entire list of segments is sent with the same optical burst header (control packet) and a short time between two segments. When contention occurs, only the contention segments of the contending burst (at the beginning) will be discarded, and the remaining segments continue their way. Using this technique, it is only necessary to indicate the number of segments in a burst instead of the burst length. The burst segments can support different classes of service, and they can be positioned in the burst considering that the parts at the end have the smallest probability of being dropped.

Another approach proposed by Vokkarane *et al*. [116, 126] to resolve the problem of contention is the combination of burst segmentation with deflection. This approach can either deflect the entire burst or deflect segments to an output port different than the intended output port, instead of dropping burst segments. As mentioned above, this proposal presents the same problems of deflection routing in terms of looping. Another limitation of this approach is that either the burst or the segments may be deflected several times (wasting network bandwidth) and traverse a longer route (increasing total processing time).

There are two approaches for ordering the contention resolution policies when segmentation and deflection are combined [116, 126]:

- *Segment-first* – if the remaining length of the original burst is shorter than the contending burst, the original burst is segmented and its tail is deflected, otherwise the contending burst is deflected (Figure 2.28);
- *Deflect-first* – the contending burst is deflected if the alternative port is free; if the alternative port is busy and the remaining length of the original burst is shorter, the original burst is segmented and its tail is dropped, otherwise, if the contending burst is shorter, the original burst is dropped.

Figure 2.28 exemplifies a contention resolution using segmentation with deflection applying the *segment-first* policy. As it may be seen in this figure, when *Burst B* arrives to the switch (contending burst), the original burst (*Burst A*) is segmented (in segments A_1 and A_2). Then, *Burst B* goes to output port 1 (*Output 1*) and segment A_2 is deflected to output port 2 (*Output 2*) as a new burst. This figure does not represent the switching configuration time.

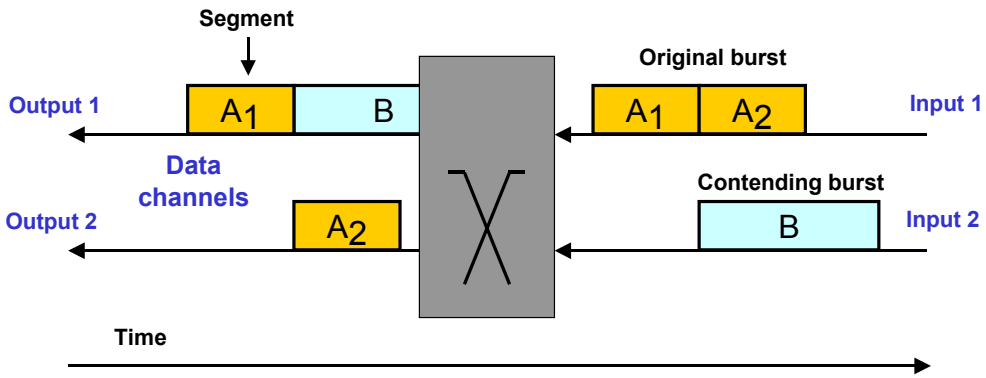


Fig. 2.28. Contention resolution using segmentation with deflection for two contending bursts.

Vokkarane and Jue in [127] proposed a new scheme based on segmentation with prioritized deflection routing in order to provide QoS differentiated services in OBS networks, considering two classes of priority (high or low) for the contending bursts. Under this scheme, bursts with high priority have lower loss and delay than bursts with low priority. Furthermore, policies that use deflection have a better performance than the ones that use limited or no deflection policies.

Detti *et al.* [132] developed a new contention resolution technique, called *burst dropping* technique, in the context of the optical composite burst switching (OCBS) [174]. Under burst dropping technique, when contention occurs, it discards only the initial part of the burst and forwards the rest of the burst, beginning at the instant in which one wavelength becomes free. These authors illustrated and analyzed the burst dropping technique and concluded that it allows an increase in the switch performance, in terms of packets loss probability, concerning the adoption of the wavelength dimension technique alone.

Table 2.6 summarizes the main advantages and disadvantages of the most relevant contention resolution schemes presented by Chen *et al.* in [63]:

Contention resolution	**Advantages**	**Disadvantages**
Fiber Delay Lines (FDLs)	Conceptually simple; mature technology; network throughput	Bulky FDLs; extra delay; more voids; additional hardware cost
Wavelength conversion	Much lower burst loss	Immature, complex, and expensive technology
Deflecting routing	No extra hardware requirement; easy to implement	Out of order arrivals; possible instability; efficient when traffic load is low; possible looping
Burst segmentation	Finer contention resolution	Complicated control

Table 2.6. Comparison of contention resolution schemes.

Several policies can be followed to resolve the contention resolution, combining one or more of the techniques presented above. Vokkarane *et al.* [93, 116] suggest five policies that can be used to handle the contention in OBS networks:

- *Drop policy* – drop the entire contending burst;

- *Deflect and drop policy* – deflect the contending burst to the alternative port and, if the port is busy, the burst is dropped;
- *Segment and drop policy* – select the contending burst; the original burst is segmented and its segmented tail is dropped;
- *Segment, deflect and drop policy* – segment the original burst, and its segmented tail can be deflected if an alternative port is free, otherwise the tail is dropped;
- *Deflect, segment and drop policy* – if there is an alternative port available the contending burst is deflected, otherwise the original burst is segmented and its tail is dropped, while the contending burst is transmitted.

2.6 Other OBS Issues

This section provides a brief description of various mechanisms related with critical issues in OBS networks, namely Quality-of-Service, TCP over OBS, and Multicasting. OBS technology demonstrates higher potential for more diverse applications. Hence, to conclude this section, a brief discussion of some of the practical applications proposed for OBS technology is included, such as Grid computing and distributed databases.

2.6.1 Quality of Service Support in OBS Networks

Quality-of-Service (QoS) in the Internet is an important issue due to service requirements needed by different applications. Internet is an IP network and current IP version provides only a best-effort service model. Under this service model, the network allocates bandwidth to all active users as best as it can and traffic is processed as quickly as possible, but there are no guarantees to the end-to-end delay or the packet loss rate [200-202]. In the context of next generation, optical Internet is critical due to service requirements needed by different applications. This is very important to support interactive, multimedia, and real-time applications (e.g. Internet telephony, videoconferencing, Computer Supported Cooperative Work, and

telescience) that require high quality of service (QoS), such as low delay, low jitter, and low loss probability with high bandwidth utilization [136, 195, 201].

QoS has been a big issue under research in recent past years. This research work has culminated in the proposal of two services architectures by Internet Engineering Task Force (IETF): integrated services (IntServ) [203] and differentiated services (DiffServ) [204, 205]. IntServ attains QoS guarantees to individual sessions. It provides services on a per flow basis where a flow is a packet stream with common source address, destination address and port number. IntServ routers must maintain per flow state information. IntServ is not scalable for the Internet size. On the other hand, with DiffServ is intended to address the scalability requirements for the global Internet. DiffServ works in the core of the network providing relative QoS advantages for different classes of traffic aggregates. In the DiffServ model, packets are classified as belonging to a flow if they have the same marking in their type of service (ToS) byte (in IPv4) or traffic-class byte (in IPv6). DiffServ architecture was selected as the model that provides QoS over the Internet.

In OBS networks QoS is also a big issue. QoS mechanisms can be implemented in conjunction with existing resource reservation protocols and contention resolution techniques. Thus, QoS in OBS networks can be classified into two models: relative QoS and absolute QoS [64, 206-208]. In the relative QoS model, traffic is classified into classes and the performance of each class is defined relatively to other classes. Under this model, the performance of each class is not defined quantitatively in absolute terms. In such methods, there is no upper bound guarantee on the high priority-class loss probability.

Concerning absolute QoS (or quantitative QoS) model, it provides a bound guarantee for the desired traffic metric such as loss probability of different classes. Typically, real-time applications with delay and bandwidth constraints, such as multimedia or videoconferencing, require such hard guarantees. For example, as mentioned in [206], for Internet service providers (ISP's), this QoS model is preferred because it is supposed and guaranteed that each user (client) receives an expected level of performance.

In this section, several QoS schemes for OBS networks are presented as examples of relative and absolute QoS models. In OBS networks, QoS differentiation can be offered at some point of the network. QoS differentiation includes differentiated offset times, differentiated burst assembly, and differentiated scheduling. Thus, in this section the respective mechanisms are mentioned that include QoS proposals.

Several relative QoS differentiation schemes have been proposed in the literature. In terms of offset-based QoS, an extra offset is given to data bursts with higher priority resulting in them having relatively lower overall loss probability. This scheme, known as prioritized JET (pJET), is proposed in [136] and its limitations are discussed in [209, 210]. However, it was shown that an offset-based scheme may not be a good solution because with larger low-priority bursts higher loss was observed than with smaller low-priority bursts [209, 210]. In [210] a proportional QoS scheme based on per-hop information is proposed. In this case, in order to maintain the differentiation loss factor between different classes, an intentional burst dropping scheme is employed. Another proposal was presented in [106]. Under this technique, a proportional bandwidth scheme is used in parallel with policing on the burst assembly mechanism and with FDL buffering. In [211], relative QoS is provided by maintaining the number of wavelengths occupied by each class of bursts. In this scheme, each class of service has a preset usage ratio of available bandwidth. Incoming bursts that are under-utilizing their share can preempt data bursts violating their assigned share.

In terms of absolute QoS schemes, in [206, 212], early dropping and wavelength grouping schemes are proposed. In the former, bursts of lower priority class are probabilistically dropped in order to guarantee the loss probability of higher priority class traffic. In the wavelength grouping scheme, the traffic is classified into different groups, and a label is assigned to each group. A minimum number of wavelengths can be provisioned for each group. An edge-to-edge resource reservation protocol is proposed in [208] in order to guarantee the edge-to-edge loss probability. In this scheme, based on the available intermediate link states, the egress node uses a class allocation algorithm to assign each intermediate link a class supporting the related burst flows.

2.6.2 TCP over OBS

The most popular applications in the Internet consist in TCP-based applications, such as World Wide Web (using hypertext transfer protocol – HTTP), electronic mail (using simple mail transfer protocol - SMTP) or grid computing. Therefore, OBS networks must support TCP and manage TCP-based applications without degrading TCP layer performance. For TCP, the effect of OBS burst loss and burst assembly, can cause a significant impact on the TCP performance.

Furthermore, due the nature of OBS core networks (using one-way resource reservation protocols and bufferless core nodes), random burst loss may occur even at low traffic load. Then, if a burst is lost, it does not necessarily designate congestion in OBS network and at higher layers (such as TCP) will need to manage the retransmission of the lost data at a later time. When a burst loss occurs, at the TCP Layer, this can be interpreted as congestion in the network and, as a consequence, unnecessarily reduces the throughput. This congestion at the TCP Layer is referred to as timeout (or false time out - FTO) by some TCP implementations, such as TCP Reno, New Reno and TCP SACK [213].

In [213], a new TCP implementation called Burst TCP (BTCP) is proposed which can detect FTOs and react properly, considering these three TCP implementations. BTCP uses three FTO detection methods based on burst length estimation, burst ACK, and burst NAK, respectively. The first method estimates the number of segments to include in the same burst. Using this method, every segment within a window is contained in a burst. If some bursts are lost, it denotes congestion in OBS networks. Using burst ACK method, when a time out occurs, the source can know whether all packets are in the same burst or not. If all packets were in the same burst, it is considered a false time out. Otherwise, it is considered a true time out. This approach is implemented in ingress edge nodes and it is necessary that they understand TCP and send a burst ACK to BTCP senders. The last approach is called burst NAK. Under this method, a burst header packet (setup message) contains information about the TCP packets contained in his correspondent data burst. When a burst is dropped, the core node where it occurs analyses the setup message and sends a burst negative acknowledgement (burst NAK) to the BTCP sender. If the BTCP sender finds that all TCP packets in a congestion window belong to the same burst lost, then it concludes that this timeout is a false timeout. Otherwise, if all TCP packets in a congestion window do not belong to the same burst lost, then the timeout is a true timeout.

Recently, several works have evaluated TCP throughput over an OBS network. The impact of data burst assembly delay on TCP over an OBS layer, using TCP Reno, has been investigated in [214]. The authors analyzed two opposite effects: the data burst assembly delay and the correlation benefit. The correlation benefit is the time correlation among the segment loss events and among the segment delivery events. They conclude that the higher correlation benefit associated with a larger number of segments that a connection aggregates inside a burst. Furthermore, the correlation

benefit is maximized for a loss probability equal to the inverse of the maximum congestion window and it disappears in the extreme values of loss probabilities.

In [215], the impact of data-burst lengths, burst-assembly times, and data burst drop rates is examined. This study suggests that for low loss probabilities, increasing burst sizes results in higher throughput and increased delay. On the other hand, for high loss probabilities, there is no significant gain with increasing burst sizes. Other studies have proposed additional features for OBS networks, such as retransmission capability or burst acknowledgment, in order to improve the TCP throughput over OBS network. One way to achieve reducing the possibility of false congestion detection by the TCP layer is to retransmit data bursts at OBS layer, as proposed in [216]. Through simulation, authors concluded that the retransmission-based OBS could significantly improve the TCP throughput over OBS.

2.6.3 Multicasting

Multicasting is an area of practical interest in networks and it is becoming very important in the Internet. Multicasting is the delivery of data from a source to multiple destinations simultaneously. Concerning optical networks, multicasting in WDM networks may be implemented using two approaches, one based on wavelength routing [217] and other based on OBS [218, 219]. In wavelength routing, multicast data will be switched to one or more outgoing wavelengths according to the incoming wavelength that carries it (called a light-tree in [217]), i.e., a wavelength needs to be dedicated to each branch of a multicast tree. Multicasting in OBS does not need a dedicated wavelength to a multicast tree. In [218] a distributed protocol is proposed, which modifies a multicast tree constructed by the IP multicast protocol *distance vector multicast routing protocol* (DVMRP) into a multicast forest where none of the multicast incapable switches is used as a branching point. This new protocol uses only local information of the WDM layer and does not requires any changes made to the IP multicasting protocol.

In [219], three schemes for multicasting in OBS networks are proposed: Separate Multicasting (S-MCAST), Multiple Unicasting (M-UCAST) and Tree-Shared Multicasting (TS-MCAST). In S-MCAST, it is implemented by sending multiple unicast bursts for the multicast destinations. In the M-UCAST approach, the multicast traffic aggregated in a burst is copied and each copy is sent as unicast traffic to the multicast destinations. In TS-MCAST there are two possibilities to implement a

multicast session, i.e., a multicast session can either have its own multicast tree or it can share a set of multicast trees. A set of multicast sessions originating from an edge node is divided into subsets, called Multicast Sharing Classes (MSCs) [219].

For TS-MCAST using MSCs, authors proposed three strategies, called Equal Coverage, Super Coverage and Overlapping Coverage, for deciding which subset of multicast sessions from the same source edge node should become a MSC. In Equal Coverage, multicast sessions with the same destination edge nodes are grouped into the same MSC. In Super Coverage, multicast sessions with a subset of the destination edge nodes are grouped into the same MSC. The most general strategy is Overlapping Coverage. Under this strategy, multicast sessions are grouped in sets that have a sufficient degree of overlap in destination edge nodes. In [220], a new strategy for TS-MCAST, called Overlapping Coverage by Maximization is proposed, to optimize the sharing gain at the ingress edge node. Under this strategy, an exhaustive search is made to obtain the best combination of subsets that can provide the maximum sharing gain at the edge node. Another proposal described in [220] is based on heuristic algorithms for managing dynamic sessions in shared multicast trees. Using these algorithms, each edge node may dynamically join and leave multicast sets. Hence, MSCs must be updated after each operation.

M-UCAST and TS-MCAST were proposed to reduce the number of guard band of bursts and correspondent control packets per unit of multicast data while S-MCAST is considered a normal approach. Authors concluded that proposed TS-MCAST scheme performs better than S-MCAST for every case while M-UCAST may perform better than S-MCAST under certain conditions [219]. To support tree-shared multicast, heuristic algorithms perform better than Overlapping Coverage by Maximization with less computational overhead [220].

Another proposal addressing the problem of reliable multicasting is presented in [221]. The authors describe a new multicast protocol that supports burst recovery in the OBS network. When a burst that belongs to a multicast session is lost, the reliable IP multicast protocols or TCP Layer need to retransmit data that belong to that burst. This implies a larger number of retransmissions and increases the end-to-end delay. Under this proposed protocol, if a burst reaches its destination, the destination edge nodes send an acknowledgement (ACK) message towards the multicast source. However, if a node drops a burst, this node sends a negative acknowledgement (NAK) message towards the multicast source. When the MAK message arrives at the first upstream node of the multicast tree, this node sends a retransmission request to the nearby multicast member node that successfully

received the burst. Thus, only one edge router member responds to the request. It is required that every node that belongs to a multicast tree maintains the state information in order to forward signaling packets (ACK or NAK) upstream along the multicast tree.

2.6.4 Burst grooming

In the literature, there are some references that are addressing data burst grooming in OBS. Burst grooming is defined as the alignment of several bursts that have the same destination closer in time and, thus, they can be switched as one unit until they are separated again [222]. Data burst grooming can be an effective scheme to improve network performance when the packet arrival rate is low and data bursts (aggregation size) must maintain a minimum length due to core node's slow switching time. In [222], it is considered that data burst grooming at core nodes where several sub-bursts sharing a common path can be aggregated together in order to reduce switching overhead. The aggregated sub-bursts can be separated at a downstream node prior to reaching their final destinations. This type of grooming is called *burst selective grooming*.

Another proposal is presented in [223], where authors address the problem of data burst grooming at the edge node and focus on improving loss probability and average end-to-end packet delay. They provide edge node architecture for enabling burst grooming and propose several data burst grooming heuristic algorithms.

2.6.5 OBS Applications

OBS technology has been considered as the underlying network technology for various applications with large data requests and sensitive to path delay. One such application is *distributed database*. A distributed database (or data warehouse) is a collection of databases located at different geographic locations and connected through a network [224]. In these networks, large pieces of data from different locations must be aggregated for computation. Hence, minimizing the delay in data aggregation is a key issue in improving the overall system throughput. In such

applications, optical burst switching technology can achieve efficient data assembly and path setup while reducing network overhead.

Another attractive area where OBS has been considered as an effective underlying technology is global Grid computing. Grid computing uses the concept of global computation network (grid) and it is understood as distributed computing, using network features and capabilities to offer efficient use of remote resources, with large bandwidth, storage, and computational requirements. A generic OBS-based architecture suitable to support Grid computing has been proposed in [225, 226] and key issues such as signaling issues, *anycast* routing, and transforming jobs into individual data bursts are discussed. Such areas are subjects of many ongoing research activities.

CD/DVD delivery is another example considered as applications in OBS networks. For CD/DVD delivery, high-speed data transmission to end users can be achieved in the OBS network where data of one CD/DVD is encapsulated into a burst [227].

The developed concepts and protocols for OBS networks are not limited to optical networks. Many of the above-mentioned techniques and models can also be extended to sensor and satellite networks. For example, sensor networks can potentially benefit from similar assembly strategies and grooming techniques developed for OBS networks. In satellite communications, where the network is less delay sensitive and has limited number of satellite switch nodes, data packets transmitted between transponders can be aggregated into data bursts with out-of-band signaling. Such networks can be more flexible and efficient than traditional SS/TDMA-based (Satellite-Switched Time Division Multiple) networks in terms of offering wideband capacity. Many of the contention resolution policies, scheduling algorithms, as well as QoS models, specifically developed for OBS networks can be potentially utilized for burst-based satellite networks.

2.7 Conclusions

This chapter reviewed the state of the art within OBS paradigm and presented it in detail taking into account its most relevant aspects. After an introduction to the topic OBS, Section 2.2 described the OBS network architecture considering both edge and core node architectures. In Section 2.3 was presented the burst assembly process and its most relevant algorithms. The Section 2.4 explained the existing resource

reservation protocols and the proposal of a new one-way resource reservation protocol, called Enhanced Just-in-Time (E-JIT). This contribution comes from the analysis of the existing one-way resource reservation protocols, taking into account their operation, implementation and relative performance (shown in Chapters 4 and 5). The contention resolution and schemes to solve the contention problem were described in Section 2.5. Section 2.6 presented other relevant OBS issues like quality of service, TCP over OBS, Multicasting, burst grooming, and OBS applications

Chapter 3

Design and Implementation of a Simulation Tool for OBS Networks

3.1 Introduction

Optical burst switching (OBS) technology raises a number of significant questions related with the analysis of the performance of different resource reservation protocols. Several network parameters may be taken into account, namely, the network size, the network topology, the number of channels per link, the number of edge nodes per core, the edge to core node delay, the propagation delay between core nodes, the burst offset length, the setup messages processing time and the optical switch configuration time. These OBS parameters may have a significant impact on the network performance. Therefore, these questions may be answered with a toll that simulates the behavior of an OBS network, given the inexistence of such networks in the real world, although there are some testbeds as reported in [75, 141-143].

Previous works in optical networks simulators are based on packet traffic (e.g. IP networks), which is significantly different from the bursty traffic in an OBS

network, since bursts are transmitted through the OBS network in a transparent way, in the sense that the network does not recognize neither the end of burst nor its content. Therefore, new tools are needed in order to include the specific features of OBS traffic at the network layer. This chapter presents a proposal of an object-oriented approach for the development of an OBS simulator, called OBSim, built in Java. OBSim supports studies to evaluate the performance of the resource reservation protocols considered in this study (JIT, JumpStart, JIT$^+$, JET, and Horizon) and the proposal of new resource reservation protocol (E-JIT).

This simulation tool is designed to implement a model of OBS networks based on objects and it was programmed in an Object-Oriented Programming (OOP) built model, with the following objectives:

- To compare the performance of different resource reservation protocols based on the burst loss probability;
- To study the influence of different network profiles on the performance of OBS networks;
- To evaluate the performance of OBS networks for different network topologies defined by the user;
- To compare OBS performance with the performance of other technologies;
- To test new OBS resource reservation protocols.

The remainder of this chapter is organized as follows. Section 3.2 analyses methodologies for performance evaluation in OBS networks. Section 3.3 presents an overview of the modeling and simulation techniques and the burst traffic model that was followed. This section includes networks topologies used in this study to evaluate the performance of the considered resource reservation protocols. Section 3.4 describes the design of OBSim simulator, including a detailed overview of its main characteristics: design considerations, simulator architecture, session traffic and scenario generations, and the input user interface of OBSim. Section 3.5 is devoted to the validation of the simulator results and the main conclusions of this chapter are presented in section 3.6.

This chapter is partially based on papers [18, 19, 28].

3.2 Methodologies for Performance Evaluation in OBS Networks

Regarding network performance, research and development tools fall into three different categories: analytical tools, *in situ* measurements, and simulators [144, 228]. All of these three techniques can be used during a network project. However, the evaluation of performance parameters may involve prototype developments and may need laboratorial experiments or *in situ* measurements. Thus, this technique is very expensive and slow, and furthermore it is not flexible. For these reasons, that technique is only used at the final stage of a project to test the robustness and the performance of the projected system. Analytical techniques are usually useful in the beginning of these kind of projects because these techniques are based on mathematical models that produce results that move far away from results obtained by measurement because optical networks are very complex. However, they are very efficient from the computational point of view. Simulation techniques allow modeling in detail of the system to analyze and they are more flexible than analytical tools in terms of different network topologies with changing number of nodes, links and interconnections. Therefore, optical networks simulators are recognized as essential tools to evaluate the performance of these systems mainly in the phase where there are several possibilities to consider or in phases that one intends to optimize and compare networks performance. However, developing optical networks requires deep knowledge not only from programming and simulation environment but also devices and sub-systems (e.g. edge and core nodes, links, reservation protocols…) that comprise the system to simulate. Furthermore, developing simulators requires long time programming and the correction of errors. On the other hand, it is possible to buy simulators if they are available commercially, but these kind of simulators with high-level detail, when available, are usually very expensive.

Concerning analytical models for performance evaluation of OBS networks, there are several proposals in the literature. These techniques are efficient from a computational point of view, but the utilization of these techniques requires very restrictive assumptions that may turn impossible a detailed description of the network [229]. Teng and Rouskas [80, 115, 133] propose analytical models to evaluate the performance of JIT, JET and Horizon, but their proposals only evaluate a single OBS node. Another analytical model to evaluate the performance of OBS networks was proposed in [230]. This model only considers JIT and JET protocols and

it presents a limitation concerning the study of low-connectivity topologies. These authors evaluate their proposal under an irregular topology but it is not possible to apply the model to low-connectivity topologies because it is assumed that each blocking event occurs independently from link to link along any route. Other analytical models for JIT, JET, TAW, and differentiated intermediate node initiated signaling were presented in [64]. These models only evaluate the end-to-end delay of each OBS signaling techniques.

Simulators may be developed using high-level language programming, such as PASCAL, C/C++, Java, or FORTRAN, or they may be developed using specific simulation language, such as MATLAB [231], Ptolemy II [232, 233] (set of Java packages developed at the University of California at Berkley), SSFNet [234, 235] or SIMSCRPT II.5 [236], among others. Some of the simulation languages possess specific modules for communications. MATLAB is one of these cases because it has a specific toolbox for communications. Furthermore, MATLAB is associated with another simulation tool, called Simulink [237]. Simulink is a platform for multidomain simulation and Model-Based Design for dynamic systems. It provides an interactive graphical environment and a customizable set of block libraries, and can be extended for specialized applications. However, running simulations using MATLAB simulators is slower than running simulations using high-level language programming simulators. Even so, this problem could be minimized through the conversion to MEX files and through the use of a C/C++ compiler. MEX-files are dynamically linked subroutines that MATLAB can automatically load and execute. They provide a mechanism by which it is possible to call C and Fortran subroutines from MATLAB as if they were built-in functions. Ptolemy II is a software framework developed as part of the Ptolemy project. It is a Java-based component assembly framework with a graphical user interface called Vergil. Vergil, itself, is a component assembly defined in Ptolemy II. The Ptolemy project studies modeling, simulation, and design of concurrent, real-time, and embedded systems. The focus is on assembly of concurrent components. Scalable Simulation Framework Network Models (SSFNet) are composed by open-source Java models of protocols (IP, TCP, User Datagram Protocol - UDP, Border Gateway Protocol version 4 - BGP4, Open Shortest Path First - OSPF, and others), network elements (hosts, routers, links, LANs), and various classes for realistic multi-protocol, multi-domain Internet modeling and simulation.

Nowadays, powerful simulators are available commercially, such as Optsim [238], LinkSIM [239], and OPNET Modeler [240]. However, these solutions do not include modules for OBS networks.

Previous works were focused on simulating traffic for several types of networks and they were primarily designed to simulate TCP/IP traffic. The main example is

ns-2 (network simulator version 2) [241], developed on C++ and based on a project started in 1989 (called Real Network Simulator [242]), which has been widely used for network protocol performance studies [243]. This network simulator is available for free from ISI [241] and it is documented in [244]. There have been also other developments in the area of simulation, such as OWns [243, 245], this simulator being an extension to the ns-2. OWns is an Optical Wavelength Division Multiplexing Network Simulator and it does not simulate OBS networks. In [246] the IND Simulation Library was found, which is an object-oriented class library for event-driven simulation implemented in C++. These classes have been designed to support the communication networks performance evaluation. While developing OBSim, one had access to the simulator developed by Teng and Rouskas from North Caroline State University, USA, and to the OBS-ns simulator released by DAWN Networking Research Lab from University of Maryland [247]. Both simulators were developed under C++ programming language. The first was not published and it has no documentation support. However, there are published results obtained by this simulator in [80, 115, 133] and these results were used to validate the results of OBSim. On the other hand, OBS-ns presents several bugs and limitations to simulate OBS networks. To solve this problem, more recently, Optical Internet Research Center (OIRC) Optical burst switching Simulator (OOS) appears as a new version of OBS-ns fixing its errors [248]. Another simulator, called NTCUns, belongs to SimReal Inc.[249], a virtual company operated by the Network and System Laboratory (NSL), Department of Computer Science and Information Engineering, National Chiao Tung University (NCTU), Taiwan. NTCUns is a network simulator capable of simulating wired and wireless IP networks, optical networks, and GPRS cellular networks. Recently, this simulator incorporated several modules to construct the protocol stacks used in OBS networks. Therefore, OBSim is a tool that also gathers contributes from all those previous work in the area of network simulation.

As above mentioned, analytical models are not a good choice because the proposals presented in the literature [64, 80, 115, 133, 230] neither cover the resource reservation protocols nor can be used for network topologies under study in this thesis. This study considers both regular and irregular topologies with different number of nodes ranging between 10 and 30 nodes. On the other hand, simulation techniques are a good resource when networks under study are *i*) very complex to analyze using analytical tools or *ii*) very expensive to investigate by measurement or by prototype developing, or by both reasons [229]. In this case, it was necessary to develop a new simulator from scratch because in the very beginning of this study, to the best of the knowledge of the author, there were no OBS simulators developed and available to use.

3.3 OBS Network Modeling

As above-mentioned, research and development tools used to study the network performance fall into three different categories: analytical tools, *in situ* measurements, and simulators [144, 228]. Therefore, OBSim is an *i*) event driven, *ii*) stochastic, and *iii*) symbolic simulator. Event driven simulators are a class of models in which data flows to the pace of events of some type; other simulators may be activity driven or time driven, namely when the simulator responds to some kind of user interaction (internal or external, user initiated or not) and the last, when the software runs at the tick of a clock [135]. In OBSim, the events that run on the simulator are messages. These may be sent by the edge nodes, or generated at a core node, as defined by the resource reservation protocols above presented.

Stochastic simulators, opposed to deterministic simulators, rely on random entities (usually random variables of numerical value) to simulate the randomness of real-life events. OBSim used the Java class Random, what generates pseudo-random values (of several types), using a congruential algorithm. Pseudo-random variables must pass two tests that certify, first, the homogeneity of its distribution, and second, the independence of the generated values [135]. Java class Random satisfies these conditions [148].

Symbolic simulators use some type of symbols to copy the behavior of real elements. In OBSim, theses symbols are Java classes, which are instantiated as needed by the software, according to the input data provided initially by the user.

3.3.1 OBS Mesh Network Under Study

In this chapter, the example of an OBS core network is used (Figure 3.1), extracted from OBS network shown in Figure 2.3, to support and present the discussed design and entities of OBSim simulator. Therefore, this sub-section shows the network topology and an example of its virtualization when an edge node sends a burst to another edge node. The network considered has 6 core nodes (the OXC and its corresponding signaling engine – or switch control unit –, numbered from 1 to 6) and 9 links (data and signaling channels are shown separately).

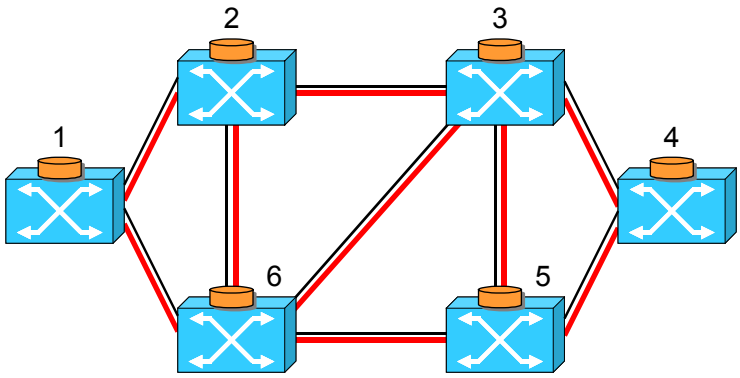


Fig. 3.1. OBS network topology with 6 nodes and 9 links.

Figure 3.2 plots a scheme and a Unified Modeling Language (UML) diagram that illustrates how a burst being sent from an edge node (edge node ‘a’ connected to core node ‘3’) to another edge node (edge node ‘b’ connected to core node ‘6’) deploys and uses a set of class instances in the simulator.

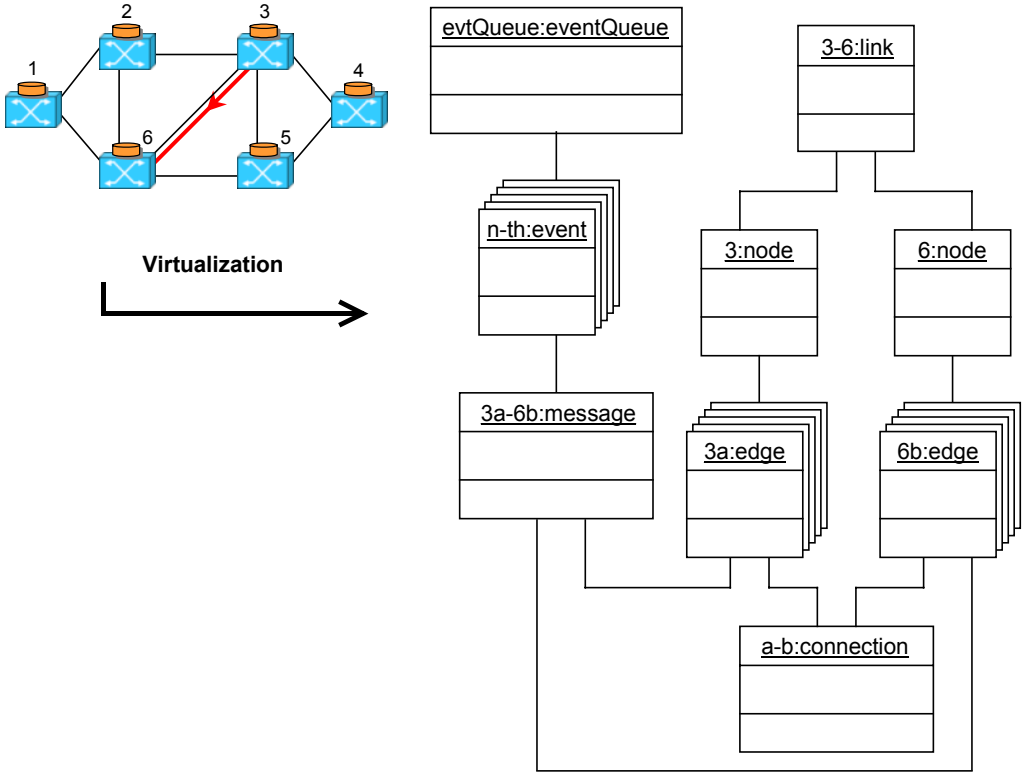


Fig. 3.2. Classes instantiated when the edge node 3a sends a burst to the edge node 6b.

3.3.2 Network Topologies for Interconnection of Core Nodes

Network topology is a pattern of links connecting pairs of nodes in a given network. A certain node is connected to one or more nodes through links. Concerning network topologies there are two types of topologies: physical and logical topologies. The physical topology is the physical structure of the network. It represents how different nodes in a network are connected to each other and the ways they communicate are determined by the topology of the network. Logical topology is the method used to pass information between nodes. It is assumed that all network topologies considered in this thesis are physical topologies and their links are considered bi-directional, i.e., the traffic flows in both senses.

Several network topologies are used to evaluate the performance assessment of the studied one-way resource reservation protocols. Mesh topologies considered in this study are both regular, like rings, chordal rings and mesh-torus, and irregular, like NSFNET, ARPANET, and European Optical Network (EON).

This sub-section describes all network topologies considered throughout this study for performance evaluation of the resource reservation protocols under study. For each network topology, the respective image with network layout is shown.

3.3.2.1 Rings and Chordal Rings

In ring topologies, each node is connected directly to two other nodes in the shape of a closed loop, one on either side of it. Figure 3.3 shows a ring topology with N=20 nodes (where N represents the number of nodes of a given topology).

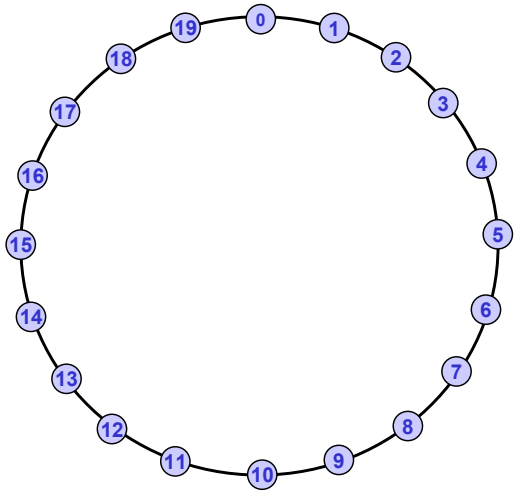


Fig. 3.3. Ring topology with N=20 nodes.

A well-known family of regular topologies with nodal degree of 3 is the chordal ring family, which was proposed by Arden and Lee [155], in the early eighties, for interconnection of multi-computer systems.

A chordal ring is basically a bi-directional ring network, in which each node has an additional bi-directional link, called a chord. The number of nodes in a chordal ring is assumed to be even, and nodes are indexed as 0, 1, 2, …, N-1 around the N-node ring. It is also assumed that each odd-numbered node i (i=1, 3,…,N-1) is connected to a node (i+w)*mod* N, where w is the chord length, which is assumed to be a positive odd. For a given number of nodes there is an optimal chord length that leads to the smallest network diameter. The network diameter is the largest among all of the shortest path lengths between all pairs of nodes, the maximum length of a path being determined by the number of hops.

In [250], a general family of degree three topologies was introduced, of which the chordal ring family is a particular case. In each node of a chordal ring, there is a link to the previous node, a link to the next node and a chord. Here, it is assumed that the links to the previous and to the next nodes are replaced by chords. Thus, each node has three chords, instead of one. Let w_1, w_2, and w_3 be the corresponding chord lengths, and N the number of nodes. A general degree three topology is represented by D3T(w_1, w_2, w_3). It is assumed that each odd-numbered node i (i=1, 3, …, N-1) is connected to the nodes (i+ w_1)*mod* N, (i+ w_2)*mod* N, and (i+ w_3)*mod* N, where the chord lengths, w_1, w_2, and w_3 are assumed to be positive odd, with $w_1 \leq N$-1, $w_2 \leq N$-1, and $w_3 \leq N$-1, and $w_i \neq w_j$, $\forall_{i \neq j} \wedge 1 \leq i,j \leq n$. In this notation, a chordal ring with chord length w_3 is simply represented by D3T(1,N-1, w_3).

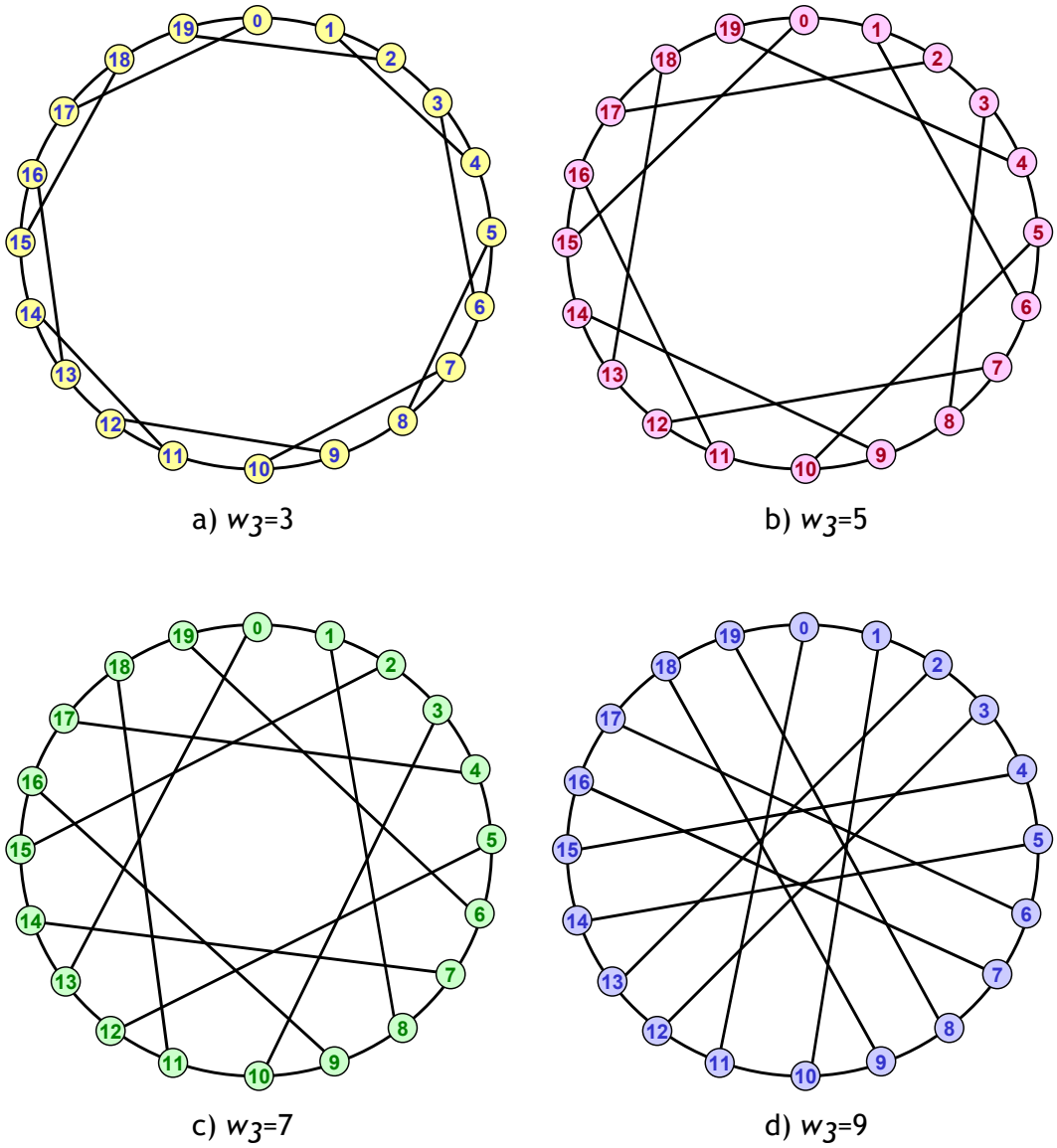
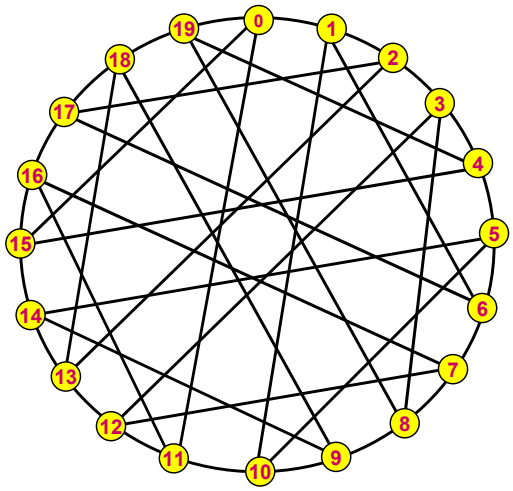


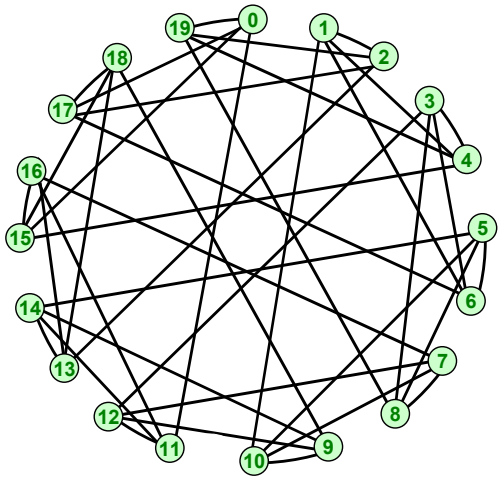
Fig. 3.4. Chordal ring networks with 20 nodes (D3T(1,19,w_3)) and chord lengths of a) w_3=3, b) w_3=5, c) w_3=7, and d) w_3=9.

In [251], a general topology for a given nodal degree was introduced. It is assumed that instead of a topology with nodal degree of 3, there is a topology with a nodal degree of *n*, where *n* is a positive integer, and instead of having 3 chords there are *n* chords. It is also assumed that each odd-numbered node *i* (*i*=1,3,…,*N*-1) is connected to the nodes $(i+w_1) mod\ N$, $(i+w_2) mod\ N$, …, $(i+w_n) mod\ N$, where the chord lengths, w_1, w_2, … w_n are assumed to be positive odds, with $w_1 \leq N-1$, $w_2 \leq N-1$, …, $w_n \leq N-1$, and $w_i \neq w_j$, $\forall_{i \neq j} \wedge 1 \leq i,j \leq n$. Now, a new notation is

presented: a general degree *n* topology is represented by DnT(w_1, w_2,…,w_n). In this new notation, a chordal ring family with three connections per node (nodal degree-three) and a chord length of w_3 is represented by D3T(1,*N*-1,w_3) and a bi-directional ring is represented by D2T(1,*N*-1). Figure 3.4 shows chordal ring networks with 20 nodes, i.e. D3T(1,19,w_3), for a) w_3=3, b) w_3=5, c) w_3=7, and d) w_3=9. As an example of degree four topologies, Figure 3.5 shows two topologies for networks with *N*=20 nodes: D4T(1,19,5,9) (degree four chordal ring with chord lengths of w_3=5 and w_4=9), and D4T(1,3,5,9).



a)



b)

Fig. 3.5. Network degree-four topologies with *N*=20 nodes: a) D4T(1,19,5,9) and b) D4T(1,3,5,9).

3.3.2.2 Mesh-torus

Another type of regular network topologies are mesh-torus. In this topology, each node is interconnected with 4 nodes and, obviously, the nodal degree is four. The 4 x 4 torus network (with *N*=16 nodes and 32 links) shown in Figure 3.6 and the 5 x 5 torus network (with *N*=25 nodes and 50 links) shown in Figure 3.7 are considered.

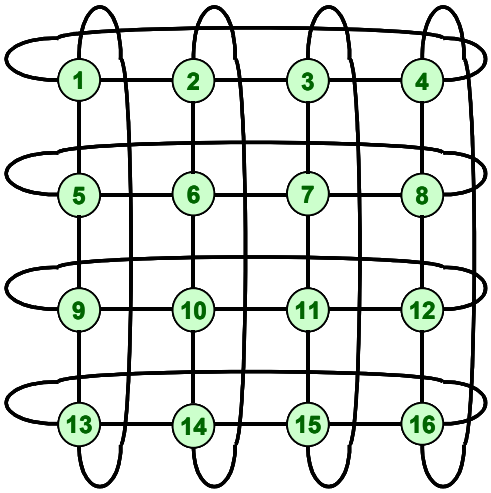


Fig. 3.6. Mesh-torus topology with *N*=16 nodes.

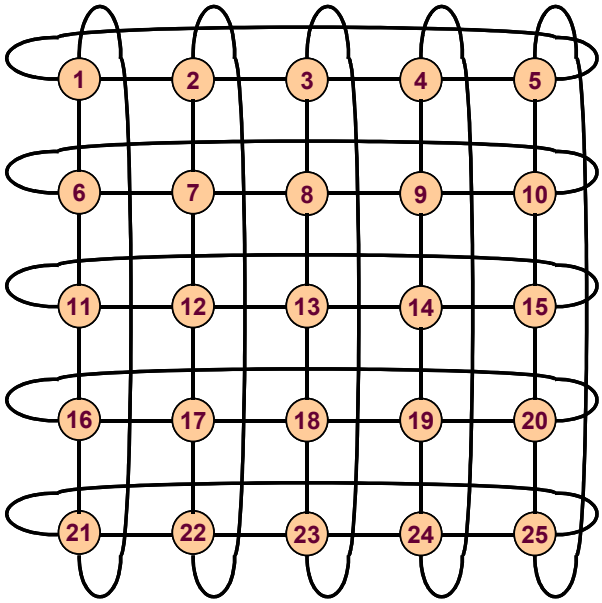


Fig. 3.7. Mesh-torus topology with *N*=25 nodes.

3.3.2.3 ARPANET

Another arbitrary network, that is considered in this study, is the 20-node ARPA network shown in Figure 3.8 [151, 153]. This network topology has *N*=20 nodes and 32 links, and its nodal degree is 3.2.

The precursor to the Internet, ARPANET was a large wide-area network created by the United States Defense Advanced Research Project Agency (ARPA). Established in 1969, ARPANET served as a test-bed for new networking technologies, linking many universities and research centers.

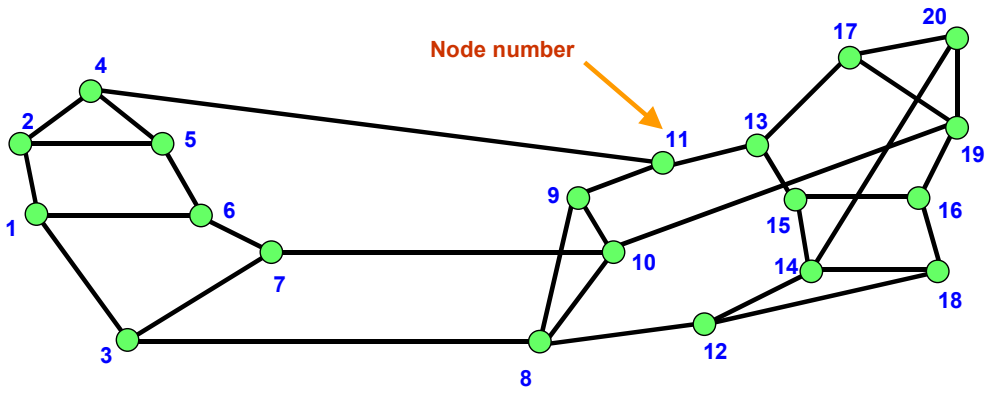


Fig. 3.8. ARPANET topology with *N*=20 nodes and 32 links.

3.3.2.4 NSFNET

NSFNET was a network developed under the auspices of the USA National Science Foundation (NSF). NSFNET replaced ARPANET as the main government network linking universities and research facilities. In 1995, however, the NSF dismantled NSFNET and replaced it with a commercial Internet backbone. At the same time, the NSF implemented a new backbone called *very high-speed Backbone Network Service* (vBNS), which serves as a testing ground for the next generation of Internet technologies.

In the research literature, one found two versions of NSFNET topology: a well-known version with *N*=14 nodes and 21 links [64, 151] (with a nodal degree of 3)

and another version with *N*=16 nodes and 25 links [152] (with a nodal degree of 3.125), presented in Figures 3.9 and 3.10, respectively. The 16-node topology is based on 14-node NSFNET and it consists in the addition of two fictitious nodes (node 1 and node 16 of Figure 3.10), to observe the effect of NSFNET's connections to Canada's communication network – CA*net [115]. Therefore, this study considers both network topologies to use more topologies of real or with real approximation networks.

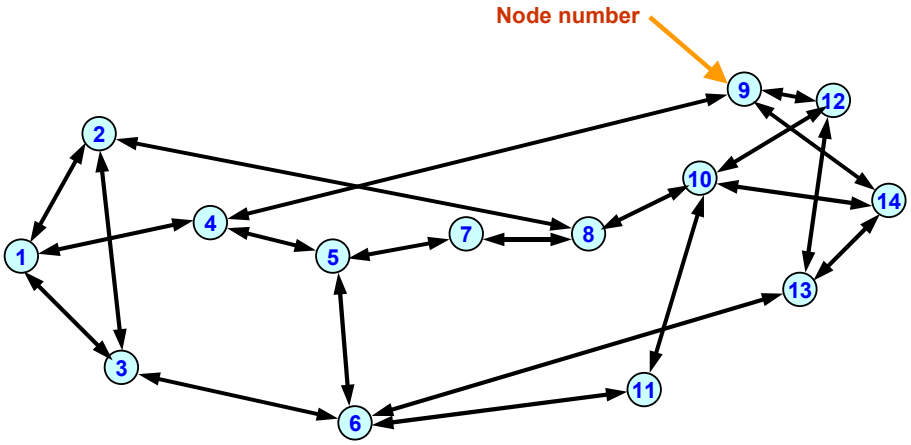


Fig. 3.9. NSFNET topology with *N*=14 nodes and 21 links.

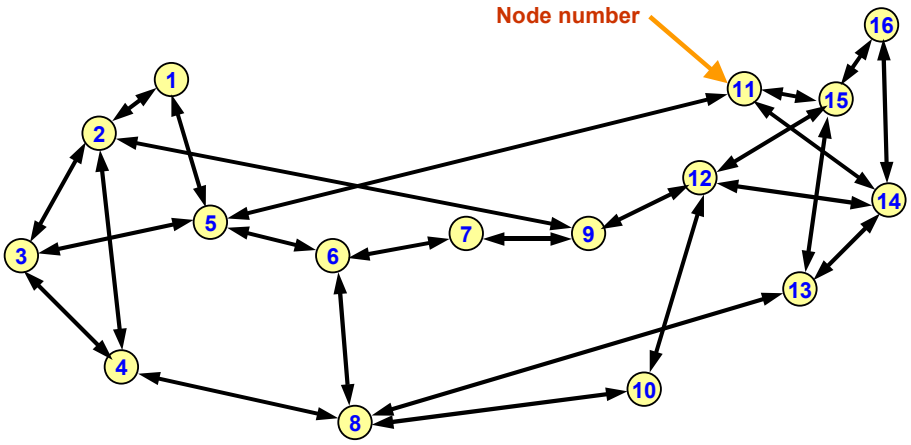


Fig. 3.10. NSFNET topology with *N*=16 nodes and 25 links.

3.3.2.5 European Optical Network

The European Optical Network (EON) was initially proposed in [252] for interconnecting the major European national capitals, taking into account reasonable assumptions about the possible traffic distributions, given the node populations and incorporating suitable structures to enhance reliability. In [154] a flat topology of the EON is presented (shown in Figure 3.11). EON design considerations were before presented in [252, 253]. Each node represents the gateway between a national network and the EON. This network is formed by a central ring that interconnects central network nodes (London, Berlin, Milan, and Paris). Inside this ring, there is a ring with four inner nodes (Brussels, Amsterdam, Prague, and Zurich), and outside, there are two outlying islands formed by groups of three nodes (Oslo, Stockholm, and Copenhagen; Rome, Athens, and Zagreb). All the remaining nodes (Lisbon, Madrid, Dublin, Vienna, and Moscow) are connected to either the central or outer ring. Thus, this network topology has *N*=19 nodes and 37 links, and its nodal degree is 3.89.

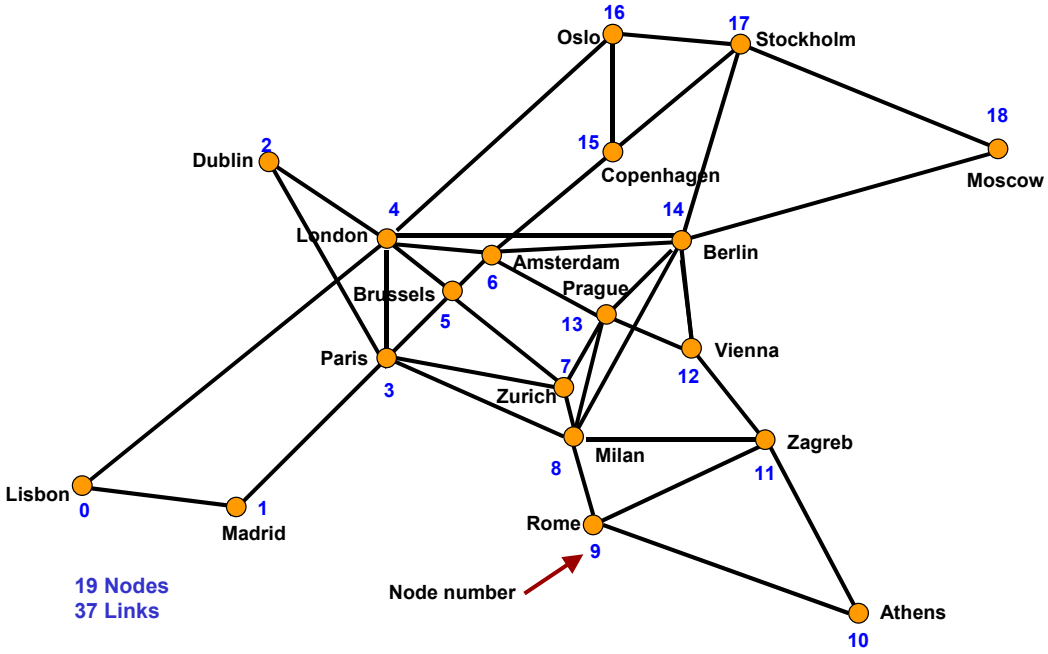


Fig. 3.11. European optical network topology with *N*=19 nodes and 37 links.

3.3.2.6 Portuguese FCCN Network

In this study, an OBS network topology is proposed (plotted in Figure 3.12) for the backbone of the Portuguese *Fundação para a Computação Científica Nacional* [156] (FCCN network), based on the network structure of the REFER Telecom SA [157]. REFER Telecom SA belongs to REFER EP [254]. Under this proposal this network is called FCCN-NET. The network interconnects the Portuguese universities, polytechnic institutes and state research labs. The choice of this number of nodes was made taking into account also the minimum number of nodes of an irregular network used in this study, such as NSFNET with *N*=14. FCCN-NET has *N*=14 nodes and 14 links – 13 nodes belong to Refer Telecom and the 14th node belongs to FCCN (Lisbon's GigaPix). It is assumed that the gateway for international traffic is the connection to the Lisbon's GigaPix, although REFER Telecom may have other international connections. In this network are considered five main nodes (Lisbon, Porto, Coimbra, Aveiro, and Lisbon's GigaPix), eight secondary nodes (Braga, Vila Real, Guarda, Covilhã, Castelo Branco, Setubal, Évora, and Faro), and an interconnection node (Entroncamento). It is assumed that the difference between main and secondary nodes is that secondary ones have half of the local traffic in terms of the probabilistic traffic load distribution. Entroncamento node was introduced in function of the REFER Telecom network infrastructure for interconnection with Lisbon, Coimbra and Castelo Branco.

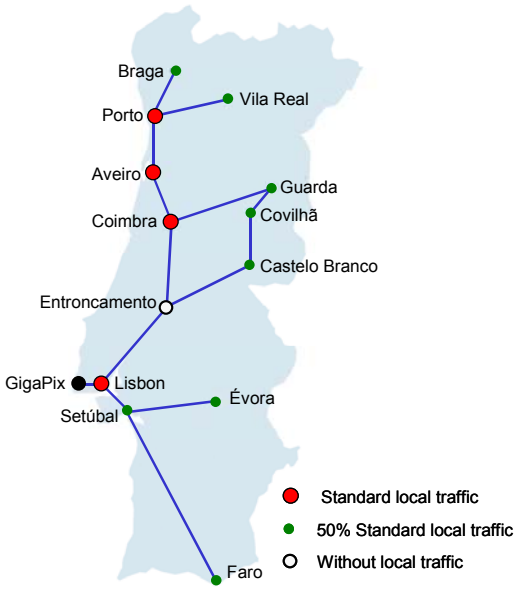


Fig. 3.12. Portuguese FCCN network topology proposed (*N*=14 nodes and 14 links).

3.3.3 Burst Traffic Model

As mentioned above (in Chapter 2), it is assumed that each OBS node requires [80, 133]: *i*) an amount of time, T_{OXC}, to configure the switch fabric of the OXC in order to set up a connection from an input port to an output port, and requires *ii*) an amount of time, $T_{Setup}(X)$ to process the setup message for the resource reservation protocol X, where X can be JIT, JumpStart, JIT$^+$, JET, horizon, and E-JIT. The offset value of a burst under protocol X, $T_{offset}(X)$ is also considered, which depends, among other factors, on the resource reservation protocol, the number of nodes the burst has already traversed, and whether the offset value is used for service differentiation. It is assumed that the number of hops in the path of a burst for a given network is uniformly distributed from the ingress core node to the egress core node, and the burst offset time ($T_{offset}(X)$) is calculated by (2.1). This study follows the simulation conditions presented in [80, 133] in terms of traffic scenarios. In obtaining the simulation results, to estimate the burst loss probability, 95% confidence intervals was estimated using the method of batch means [134, 135]. The number of batches is 30 (the minimum value to obtain the confidence interval) and each batch run lasting until at least 2560000 bursts are transmitted, assuming that each edge transmits, at least, 200 bursts and each batch contains 10 observations. The value of 2560000 transmitted bursts, at least, is obtained for a network with N=20 nodes, i.e., 200 bursts transmitted x 64 edge nodes x 20 nodes x 10 observations = 2560000 transmitted bursts. It was found that the confidence intervals are very narrow. Therefore, in this thesis, the confidence intervals are not shown in the figures, with the objective of increasing readability.

Concerning the number of data channels available per link, it is assumed that the number of channels connecting two nodes is F+1, where F is the number of data channels available in that link and the other channel represents the signaling channel. In this study, F may have the value of 16, 32, 64 or 128 data channels per link ($F=2^n$, with $4 \leq n \leq 7$). Each channel is bi-directional.

In terms of setup message arrival process (and in consequence data bursts), a Poisson point process with rate λ is assumed (where $1/\lambda$ is the mean duration of the burst interarrival time), such as in [1, 3-6, 80, 115, 133, 136]. As in [80, 93, 94, 115, 116, 137, 138], it is assumed that burst length, whether short or long, is limited [43] and follows an exponential distribution with an average burst length of $1/\mu$.

Therefore, in OBSim, taking into account the average of burst length distribution ($1/\mu$) and the setup message arrival rate λ, the burst generation ratio is represented by λ/μ. It is also assumed that bursts are sent uniformly to every core node in the network, with the exception that a core node cannot send messages to itself and, one core node may generate, at the most, one message per time-slot or time period. Edge nodes are responsible for the burst generation process, i.e., neither the ingress core node nor any other core node processes the burst.

Another important issue related with this study is related with the number of wavelength converters in each node. It is assumed that each OBS node supports full-optical wavelength conversion. In terms of number of edge nodes connected to each core node, it is assumed that, for simulation effects, they are uniformly distributed and each core connects 64 edge nodes. Between core nodes it is assumed that the geographical size is large so the typical link delays are in the order of 10ms [139].

Other parameters included in this simulation tool are the propagation delay between edge and core nodes and between core nodes. The former refers to the propagation time delay when the message (and bursts) is in transit between the edge node and the ingress/egress core node. This parameter may be introduced in the input user interface (Sub-section 3.4.4). The latter is the propagation delay between core nodes. This parameter is defined in the text file with the definition of a network topology (Sub-section 3.4.3).

In the simulation, to select a free channel for an incoming burst (with equal probability), for JIT, JumpStart, and JIT$^+$, the *random* wavelength assignment policy is used [140], where as, for JET and horizon, the latest available unused channel (LAUC) algorithm is used [95] (presented in Sub-section 2.4.4).

3.4 Design of OBS Simulator

To study the problem and the characteristics of burst traffic in OBS networks, it is necessary to evaluate the performance of different resource reservation protocols. This is achieved by studying its performance and behavior under different traffic conditions and network topologies. This tool simulates the behavior of a custom OBS network defined by the user. The simulator, called OBSim, allows to assess and

compare the performance of one-way resource reservation protocols and load profiles to a given network topology.

3.4.1 Design Considerations

As a simulator independent of existing network data encapsulation protocols was needed, the OBSim was built from scratch. Java was the programming language chosen to build OBSim for several reasons, namely:

1. The quality and ease of use of Java available programming tools;
2. The robustness of Java in object and memory handling; and
3. The wide platform portability of the code.

Several assumptions were made while building OBSim. These assumptions occur in respect of the definition of the resource reservation protocols. Concerning network modeling, these are the following:

1. All the nodes work in an independent and similar way;
2. All time scales are normalized in time-slots;
3. A path is used by a burst or by a setup message, independently of the state of the network;
4. When bursts or setup messages arrive to a node, they follow a predefined path calculated previously by Dijkstra's algorithm [255];
5. Between two consecutive nodes, the wavelength (data channel) used is chosen by the algorithm defined by the user in the network topology definition file (*random* or *first-free*) [256].

OBSim maintains an event queue that accepts and removes events, and forwards each event to its corresponding object (that with other objects compose the virtual network) so that it can be processed.

Abstraction in OBSim is achieved by the behavior of the objects of the model. The OBS network intended to simulate, is a set of defined real-world objects (eventually, real-world objects yet to be real-objects), each having its function and behavior, each interacting with the remaining objects of the network according to a defined set of rules (e.g., the algorithms of the signaling engines). As an example, the network topology defined initially by the user is processed according to the

Dijkstra's Algorithm, and for each pair of nodes, a route is defined. Figure 3.13 shows the routes found by the algorithm.

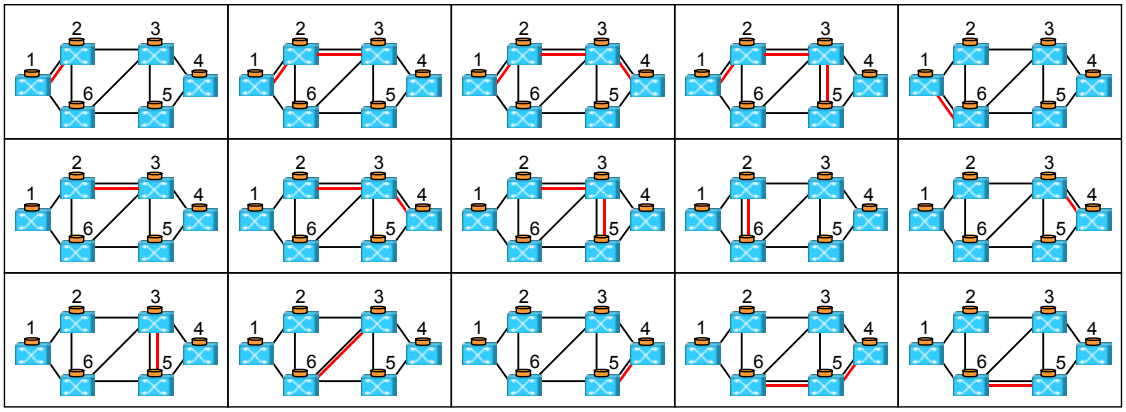


Fig. 3.13. The fifteen defined routes for topology presented in Figure 3.1.

3.4.2 Architecture of the OBSim Simulator

To simulate an OBS network, several objects (Java classes) were defined, each having methods that may be activated by other objects. These mechanisms, common in Object Oriented Programming (OOP) languages such as Java and C++, are, namely, inheritance, polymorphism, encapsulation, and also, dynamic instantiation and dynamic memory management. They allow the creation of a working model that behaves like the real OBS network would. Time flow is simulated through a queue of events, ruled by a clock, and this is presented below.

Figures 3.2 and 3.15 show how a potentially existing OBS network is virtualized in Java classes. As shown, each link is composed of two nodes, and a node may belong to more than one link (e.g., node '3').

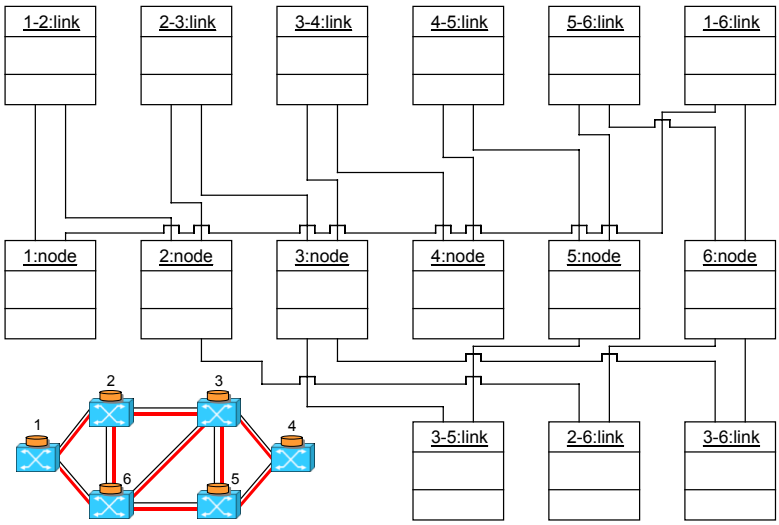


Fig. 3.14. UML object diagram modeling an OBS network (presented in Figure 3.13).

Along with node and link, other classes were created to model the behavior of an OBS network. Figure 3.15 shows the UML class diagram with the most important classes of the simulator.

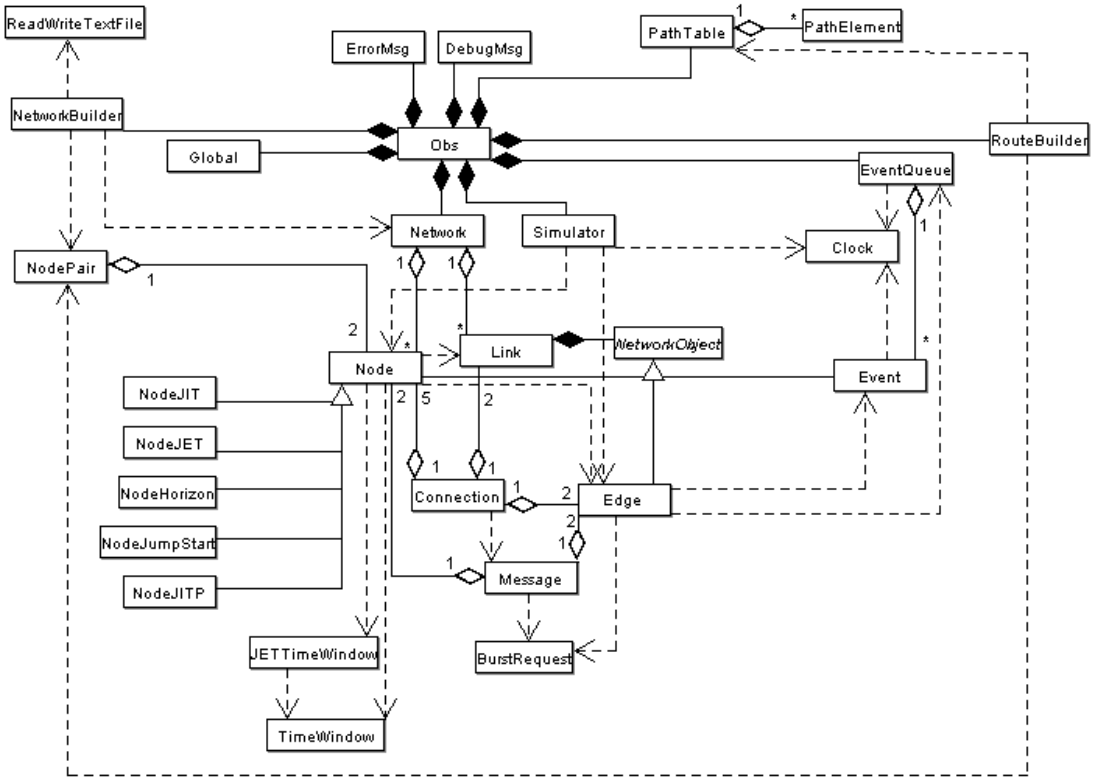


Fig. 3.15. UML class diagram for OBSim [18].

The main method is defined in *Obs class* and this calls several other objects, namely, the following:

- *NetworkFactory*, that builds the network from the topology file; this process is the virtualization that builds the class Network;
- *Network*, composed by Links and Nodes;
- *RouteBuilder*, that builds a route for any two nodes, and stores these routes in the *PathTable*; this class implements the Dijkstra's algorithm [255, 257];
- *PathTable* manages the paths defined by *RouteBuilder*;
- *Global* stores and manages the global constants and variables of the simulator;
- *Simulator*, which starts the burst request actions from each edge node at every core node;
- *EventQueue* manages all the *Events* according to the *Clock* class;
- And the classes *ErrorMsg* and *DebugMsg*, which manage the output of the debugging and error messages.

It may also be seen that the Node class is a generalization for classes *NodeJIT*, *NodeJET*, *NodeHorizon*, *NodeJumpStart*, and *NodeJITP*, which in turn, model the nodes for these resource reservation protocols. With this approach, adding a new protocol can be made by the definition of a new class and implementing it with its own set of specific algorithms.

NetworkObject is the generalization class for *Node*, *Edge* and *Event*. These are the main actors of an OBS network, and their behavior and interaction, as seen before, creates instances of classes like *Link*, *Connection*, *Event* and *Message*. In a non-parallel approach, classes *NetworkBuilder*, *Network*, *Simulator*, *EventQueue* (and *Clock*), *RouteBuilder* and *PathTable*, along with classes *ErrorMsg* and *DebugMsg* are instantiated only once. Classes *Node*, *Link* and *User* are instantiated as many times as defined in the network topology file. Other classes are instantiated as many times as needed, either by the stochastic workflow of the simulator, or by running algorithms like Dijkstra.

Another important issue that may be described at this point of the explanation of OBSim simulator is the program execution. OBSim follows a linear

functional execution scheme and performs five main tasks that are the following: *i*) to validate initial parameters defined by the user; *ii*) to read network topology from network file; *iii*) to build the network model for that network topology; *iv*) to simulate traffic; and *v*) to show simulation results.

Validation of initial parameters and the network topology allows the definition of the abstract model of a given topology to simulate. Each node of this network, according to their traffic load parameters, is instantiated to generate messages that are deposited in a queue. After reading network topology from the correspondent file the abstract model of that network is created. To create this network model the corresponding classes available on the simulator are instantiated and the correspondent data structures of each one are initialized. Next, `Routing` class executes Dijkstra algorithm for each node of the network. Each shortest path is saved in `Routing` class into a map. `Simulator` class initializes each core node and for each one, starts their edge nodes (only for each edge that transmits bursts). To start each edge node, coded in `Edge` class, a burst request is added in the event calendar (`EventCalendar` class). After all edge nodes that have permission to send bursts, place their request of a burst, OBSim will run all events that are pending in the event calendar. This event running, that may be executed either in an instance of `Node` class or in an instance of `Edge` class, generates more events that will be added to the queue. Simulation finishes when there are no more events to perform in the queue. At this point, the method that calculates ratios in function of values stocked into several classes is called. Briefly, these values may be the number of bursts sent and lost in each core node and in each edge node, and the number of bursts sent and lost in each hop either per core node or per edge node (`Node` and `Edge` classes). Using these values, the burst loss probability per hop is calculated and is shown for the user.

3.4.3 Session Traffic and Scenario Generations

Traffic generation is an important issue in the model. As OBSim is an event driven simulator, initially it is necessary to simulate the need to transmit bursts between nodes. As seen before, in a simulation, it is assumed that the bursts are sent evenly to every node in the network. Since every burst must be preceded by a setup message, and since edge nodes connected to core nodes send bursts at a

random time, it is considered that time between these messages follows an exponential distribution, simple or with an offset [146, 147]. The traffic is then simulated when the OBSim starts to process the event queue, which was, at the start of the program, loaded with requests (messages) from the edge nodes. These requests, when processed, normally generate more messages that are added to the queue. Each time a message is added to the queue, the simulator timer generates a time interval according with the distribution defined by the user, and this time is added to the simulator clock, defining the time the event will be scheduled to happen.

The network scenario is read from a text file that defines the number of nodes (core nodes and number of edge nodes per core), the number of connections, and the definition of the existing links in the network.

The text file that characterizes the topology of OBS network considered in this chapter (Figure 3.1) is illustrated in Figure 3.16. Therefore, the next paragraphs present, in detail, the contents of this file.

The first line of the text file defines the number of core nodes (6), the channel schedule algorithm used to assign the burst to a data channel on the outgoing link, and the number of links (9). For the channel schedule algorithm, one may use *0* or *1*, where *0* represents the use of Random allocation channel algorithm and *1* represents the use of First Free channel algorithm. Next, data are inserted to define the parameters of each node:

- The first two numbers represent the Coordinates (*x* and *y*) of each core node in terms of two-dimensional localization;
- The third number represents that:
  - *0* – edge nodes of this core node cannot generate burts;
  - *1* – half of edge nodes of this core node can generate bursts;
  - *2* – all edge nodes of this core node can generate bursts.
- The last number indicates the Number of Edge Nodes per Core Node.

The following lines have three values each (*1 2 10*, for the example considered). The first two values identify the number of core nodes interconnected and the last number indicates the propagation delay of this link. This value may be defined in function of the link length, i.e., in function of the real distance between the two nodes. In this study, the assumed geographical size is large so that typical link delay is in the order of 10ms [139].

```
6 0 9
0 1 2 64
1 2 2 64
2 2 2 64
3 1 2 64
2 0 2 64
1 0 2 64
1 2 10
1 6 10
2 3 10
2 6 10
3 4 10
3 5 10
3 6 10
4 5 10
5 6 10
```

Fig. 3.16. Text file with the definition of a network topology.

The creation of the network abstraction – the network model that supports the simulation – is accomplished by the classes defined in the program. Figure 3.15 is an UML diagram that partially illustrates the abstraction programmed in OBSim, for the example network showed in Figure 3.1. Figure 3.15 shows the most important classes in the simulator. These classes are responsible for the virtualization of the model.

3.4.4 Input User Interface of OBSim

The input user interface of the simulator allows the definition of several simulation attributes. The network is fully defined with these attributes and the attributes described for each node and each link in the above-mentioned text file.

The interface shown in Figure 3.17 shows the parameters used to configure the model of the OBS network. These parameters are the following:

9. *Resource reservation protocol* - This field registers the definition of the OBS resource reservation protocol to be used. The allowed values are JIT, JumpStart, JIT$^+$, JET, Horizon, and E-JIT.
10. *Generation distribution function* - This field defines the statistical model to generate the time interval between events in the simulator. The allowed functions are the Exponential and Two State Exponential.
11. *Burst generation ratio* - This field admits a real value between 0 and 1, and it represents the burst generation ratio per node (λ/μ).
12. *Available data channels per link* - This field allows numerical values between 1 and 2147483647[9] and represents the available number of data channels per link.
13. *Setup message process time* - This field admits a decimal value greater than 0, and represents the amount of time that the OXC spends to process the setup message, defined as T_{Setup} (X).
14. *Switch configuration time* - This field admits a decimal number greater than 0 and means the time the OXC takes to configure the optical switch matrix, after the setup message has been received and interpreted, defined as T_{OXC}.
15. *Edge to core node delay* - This is a numerical field greater than 0 and stores the time the message (and bursts) are in transit between the edge node and the ingress/egress core node.
16. *Network topology file* - This field stores the name of the file where the network is defined. It has an associated button (*file...*) that allows browsing the directory structure showing the available files.

[9] Largest allowed value for an `int` (integer) in Java.

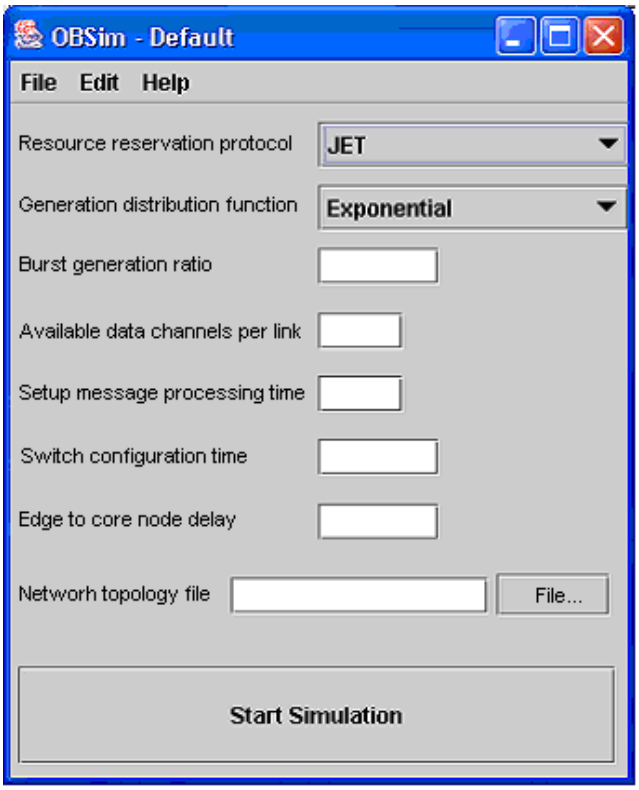


Fig. 3.17. Input user interface of OBSim.

3.5 Simulator Validation

Validation is a key issue to entrust the use of the results given by any simulator. Perros [135] defines the validation of the model as the verification of the following five steps:

1. Check the pseudo-random numbers generator
2. Check the stochastic variable generator
3. Check the logic of the simulation program
4. Relationship validity
5. Output validity

In OBSim, the accuracy of the pseudo-random number generator is guaranteed by the Java language definition standards, and confirmed through the Qui-Squared test, and the Independence test performed on the Java class Random [135, 148].

The stochastic variable generators have been separately validated by [147, 149, 150].

The logic of the simulator and the validity of the relationships are inherent to the design of the resource reservation protocols, and to the Java programming environment, above mentioned.

The output validity has been achieved through comparison with the results of [80]. For this purpose, a sample simulation was run considering a single OBS node, in isolation, for JIT, JET, and Horizon resource reservation protocols. It is assumed that [80]: T_{Setup}(JIT)=12.5μs, T_{Setup}(JET)=50μs, T_{Setup}(Horizon)=25μs, T_{OXC}=10ms, the average of burst length $1/\mu$ was set to 50ms (equal to $5T_{OXC}$), and burst arrival rate λ of *setup messages*, is such that λ/μ=32, assuming 64 edge nodes per core node.

Figure 3.18 shows the burst blocking probability as a function of the number of data channels per link for the OBS network presented above, given by OBSim and compared to the results presented in [80]. As may be seen in this figure, the results obtained by OBSim are in a close range of those published by [80]. The small variation perceived is expectable because of the stochastic nature of the events that are modeled.

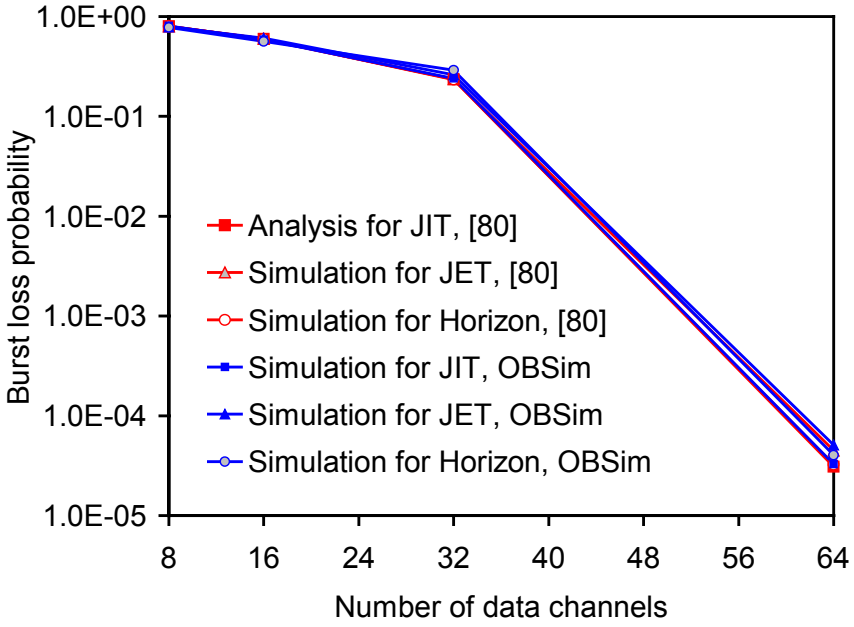


Fig. 3.18. Burst loss probability, as a function of number of data channels per link (*F*) [18], in a single OBS node for JIT, JET and Horizon resource reservation protocols given by OBSim compared to results published in [80].

3.6 Conclusions

In this chapter, the objectives, design, implementation and validation of a simulator for OBS networks, named OBSim was presented. After having studied the existing methodologies for performance evaluation of OBS networks (mainly, analytical models and simulators), the need of developing a new simulator from scratch was established. This simulator implements a model of OBS networks based on objects, which allows the estimation of the performance of a given OBS network defined by the user. The results of the simulator have been validated, and thus the simulator may be used as a tool to estimate the performance of the OBS networks. Therefore, OBSim is used as a simulation tool to support the studies of performance assessment presented in the next chapters.

Each mesh network topologies studied in this thesis, both regulars and irregulars, were also presented. Regular topologies considered are rings, chordal-rings (with nodal degree three, four, five, and six), and mesh-torus (with N=16 nodes and 32 links, and with N=25 nodes and 50 links). In terms of irregular topologies, the following are considered: NSFNET (with N=14 nodes and N=16 nodes), ARPANET, European Optical Network, and the proposal of the backbone for Portuguese FCCN (FCCN-NET).

Chapter 4

Performance Assessment of OBS Ring and Chordal Ring Networks for Existing One-way Resource Reservation Protocols

4.1 Introduction

This chapter presents a performance assessment of JIT, JumpStart, JIT$^+$, JET, and Horizon one-way resource reservation protocols in OBS networks with ring and chordal ring topologies. This performance evaluation is based on the burst loss probability obtained through simulation runs using the simulator described in Chapter 3. In this chapter, the performance metrics used to evaluate the network performance, is described followed by the study of the nodal degree and chord length gains, and the influence of the setup message processing time and OXC configuration time.

In the research literature there are several network performance studies in OBS networks, namely, in [4, 80, 113, 133, 137, 158-162]. Thus, this thesis considers bufferless networks (e.g. core nodes with no fiber delay lines), and the study focus

on the loss of data bursts in OBS mesh networks. Hence, when a burst cannot be switched it is dropped.

In [80], Teng and Rouskas considered two scenarios for simulation: one scenario is based on current available OXC and hardware processing technology, and the second is a projection for a near future considering the next 3 to 5 years. For current available technology, using the existing Micro-Electro-Mechanical Systems (MEMS) switches [163], the time to configure the OXC is T_{OXC}=10ms, and the time to process the setup messages using JITPAC controllers [141] is $T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT^{+})$=12.5μs. For JET and Horizon, to the best of the author's knowledge, they have not been implemented in hardware. Therefore, in this study, the values estimated by Teng and Rouskas [80] are assumed, i.e., the value of setup message processing time for JET is four times of $T_{Setup}(JIT)$ and for Horizon is two times of $T_{Setup}(JIT)$. Thus, $T_{Setup}(JET)=4*T_{Setup}(JIT)$=50μs and $T_{Setup}(Horizon)=2*T_{Setup}(JIT)$=25μs.

For the near future scenario, those authors present a projection considering for time to configure the OXC an improvement of three orders of magnitude (T_{OXC}=20μs), and for setup message processing time an improvement of one order of magnitude ($T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT^{+})$=1μs; $T_{Setup}(JET)$=4μs; $T_{Setup}(Horizon)$=2μs). These values are proposed assuming that the less mature OXC technology will improve faster than the more mature hardware processing technology [80]. The range of changing of these values between the current and the near future scenarios is helpful to study the effect of these times (T_{OXC} and T_{Setup}) on the performance of OBS networks.

For current scenario, it is assumed that the average of burst length distribution is $1/\mu=5*T_{OXC}$=50ms, and the setup message arrival rate, λ, is such that λ/μ=32. For the near future scenario, it is also assumed that the average of burst length distribution is $1/\mu=5*T_{OXC}$=100μs, and the setup messages arrival rate λ, is such that λ/μ=32. However, in several approaches the study is made through the variation of the value of λ/μ.

It is assumed that the value of the edge to core node delay is small, i.e., 0.5μs, and the propagation delay between core modes is equal to 10ms [139]. For channel scheduling the use of the Random channel scheduling algorithm is considered.

The remainder of this chapter is organized as follows. Section 4.2 addresses performance metrics used to assess the network performance and quantify the benefits due to the increase of nodal degree, and to quantify the benefits due to a

better choice of the chord length. Section 4.3 presents the performance comparison of OBS networks with ring and degree-three chordal ring networks, and Section 4.4 focuses on degree-four chordal ring networks. Main conclusions are presented in Section 4.5.

This chapter is partially based on papers [20, 22, 32].

4.2 Performance Metrics

This section describes three performance metrics to evaluate the network performance that are the following: burst loss probability, nodal degree gain and chord length gain. Burst loss probability is a very important performance metric to evaluate the performance of OBS networks [60]. Burst loss probability is defined as the probability that the transmitted burst does not reach its destination. In OBS networks, during burst transmission along a path, from source edge node to destination edge node, a burst may be dropped for several reasons. If control packets (setup messages) cannot reserve resources at any of the intermediate core nodes, and if the control channel itself suffers from congestion or another failure, the corresponding burst is lost.

In order to quantify the benefits due to the increase of nodal degree, and to quantify the benefits due to a better choice of the chord length, two new performance metrics are introduced: the nodal degree gain, $G_{n,k}(i,j)$, and the chord length gain, $G_{cl}(i,j; w_3,w_3^*)$.

The nodal degree gain, $G_{n,k}(i,j)$, is defined as:

$$G_{n,k}(i,j) = \frac{P_i(n)}{P_j(k)} \tag{4.1}$$

where $P_i(n)$ is the burst loss probability in the i-th hop of a degree n topology ($P_i(n)= P_i(\text{DnT}\ (w_1, w_2, \ldots, w_n))$) and $P_j(k)$ is the burst loss probability in the j-th hop of a degree k topology, for the same network conditions (same number of data channels per link - F, same number of nodes - N, same generation distribution function, same λ/μ, same setup message processing time - T_{Setup}, same time to configure the OXC - T_{OXC}, same number of edge nodes per core, same edge to core node delay, and same core to core node propagation delay), and for the same resource reservation protocol. In equation (4.1), $G_{n,k}$ is used, where n represents the

nodal-degree of the burst loss probability in the *i*-th hop - $P_i(n)$ and *k* represents the nodal-degree of the burst loss probability in the *j*-th hop - $P_j(k)$.

In order to quantify the benefits due to the choice of a chord length w_n^* instead of w_n in a chordal ring network with DnT(w_1, w_2, …, w_n), the chord length gain, $G_{cl}(i,j; w_1, \ldots, w_n^*)$, was introduced defined as:

$$G_{cl}(i,j;w_n,w_n^*) = \frac{P_i(DnT(w_1,w_2,\ldots,w_n))}{P_j(DnT(w_1,w_2,\ldots,w_n^*))} \quad (4.2)$$

where P_i(DnT(w_1, w_2, …, w_n)) is the burst loss probability for the *i*-th hop of the DnT(w_1, w_2, …, w_n), and P_j(DnT(w_1, w_2, …, w_n^*)) is the burst loss probability for the *j*-th hop of the DnT(w_1, w_2, …, w_n^*), for the same network conditions and for the same resource reservation protocol. Since it is necessary to fix one value to w_3, the first non-degenerate value for w_3 was chosen for simplicity, without loss of generality, and independent of the number of nodes, i.e., w_3=3 for the evaluation of the $G_{cl}(i,j; 3,w_3^*)$ where *i* and *j* represent the last hop of each topology.

4.3 Performance Comparison of OBS Ring and Degree-Three Chordal Ring Networks

This section focuses on the study of the performance assessment of OBS networks with ring and degree-three chordal ring topologies. In chordal ring topologies, different chord lengths can lead to different network diameters, and, therefore, to a different maximum number of hops. One interesting result that was found is concerned with the diameters of the D3T(w_1,w_2,w_3) families, for which w_2=(w_1+2) *mod N* or w_2=(w_1-2) *mod N*, where *N* represents the number of nodes. Each family of this kind, i.e. D3T(w_1, (w_1+2) *mod N*, w_3) or D3T(w_1, (w_1-2) *mod N*, w_3), with $1 \leq w_1 \leq 19$ and $w_1 \neq w_2 \neq w_3$, has a diameter which is a shifted version (with respect to w_3) of the diameter of the chordal ring family (D3T(1,*N*-1,w_3)). For this reason, the analysis is focused on chordal ring networks, i.e. D3T(1,19,w_3). Figure 4.1 shows the diameters for D3T(1,3,*w*3), D3T(1,19,*w*3), D3T(3,5,*w*3), and D3T(5,7,*w*3), which illustrates this situation. Note that the diameter of the D3T(3,1,*w*3) is the same as the diameter of D3T(1,3,*w*3) and the

diameter of the D3T(3,5,*w*3) is the same as the diameter of D3T(5,3,*w*3), i.e., in chordal rings, the order of chord length is commutative.

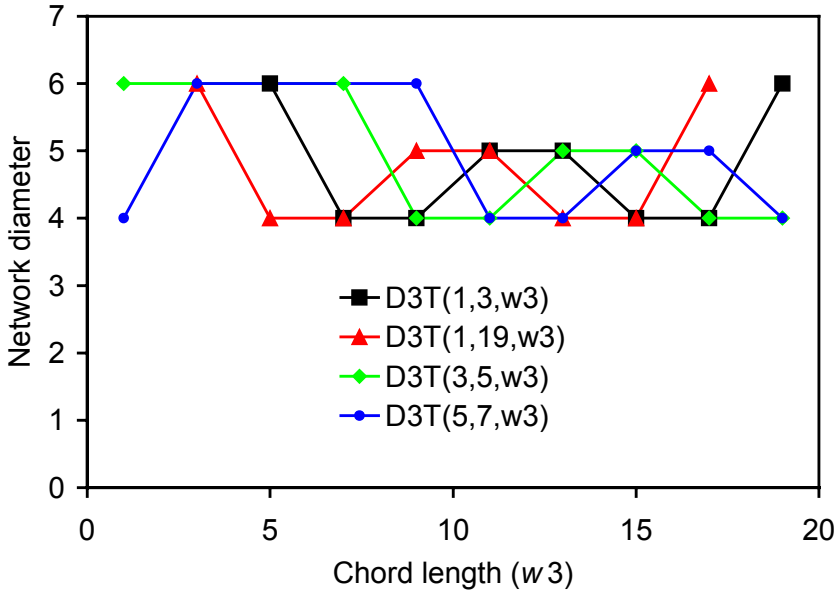


Fig. 4.1. Network diameter for D3T(1,3,w_3), D3T(1,19,w_3), D3T(3,5,w_3), and D3T(5,7,w_3), with $1 \leq w3 \leq 19$ and $w1 \neq w2 \neq w3$.

4.3.1 Study of the Nodal Degree Gain

This sub-section focuses on the study of performance assessment of OBS networks with ring and degree-three chordal ring topologies with N=20 nodes. Chordal ring topologies with 20 nodes were chosen because this number of nodes is a good approximation to real mesh network topologies, such as ARPANET (with N=20 nodes) or European Optical Network (with N=19 nodes). In this sub-section, several parameters are considered in order to study their influence on the performance of OBS networks. These parameters are the following: the number of data channels per link, the number of hops, the chord length, the burst generation ratio (λ/μ), the setup message processing time (T_{Setup}), and the OXC configuration time (T_{OXC}).

Figure 4.2 shows the burst loss probability in the first and last hops of D2T(1,19) and D3T(1,19,w_3), with w_3 equal to 3, 5, 7, and 9, for the JIT resource reservation protocol. As may be seen in this figure, chordal rings clearly have better performance than rings, where the smallest loss probability was obtained for

topologies with the smallest network diameter, i.e. D3T(1,19,5) and D3T(1,19,7). The network diameter is the least number of the maximum number of hops needed to establish a connection between one node and each other in the network. This observation may also be seen in Figures 4.3, 4.4, 4.5, and 4.6 for JumpStart, JIT$^+$, JET and Horizon, respectively. It was also observed that, in the last hop of each topology, the D3T(1,19,7) topology has the smallest burst loss probability for all resource reservation protocols. For each D2T(1,19) and D3T(1,19,w_3) topologies, the burst loss probabilities obtained for JIT, JumpStart, JIT$^+$, JET and Horizon are very similar, as may be seen in Figure 4.7, for the cases of D3T(1,19,3) and D3T(1,19,5), and in Figure 4.8, for the cases of D3T(1,19,7) and D3T(1,19,9). These figures also confirm the effect of the network diameter in the performance of each topology. The network diameter of D3T(1,19,3) and D3T(1,19,9) is equal to six and five, respectively. The network diameter of both D3T(1,19, 5) and D3T(1,19,7) is equal to four and, as may be seen, their performance is very close and better than other topologies with higher network diameter. For a given family of degree-three chordal ring topology with the same number of nodes, it is possible to conclude from the figures above-mentioned that topologies with lower network diameter have the best performance.

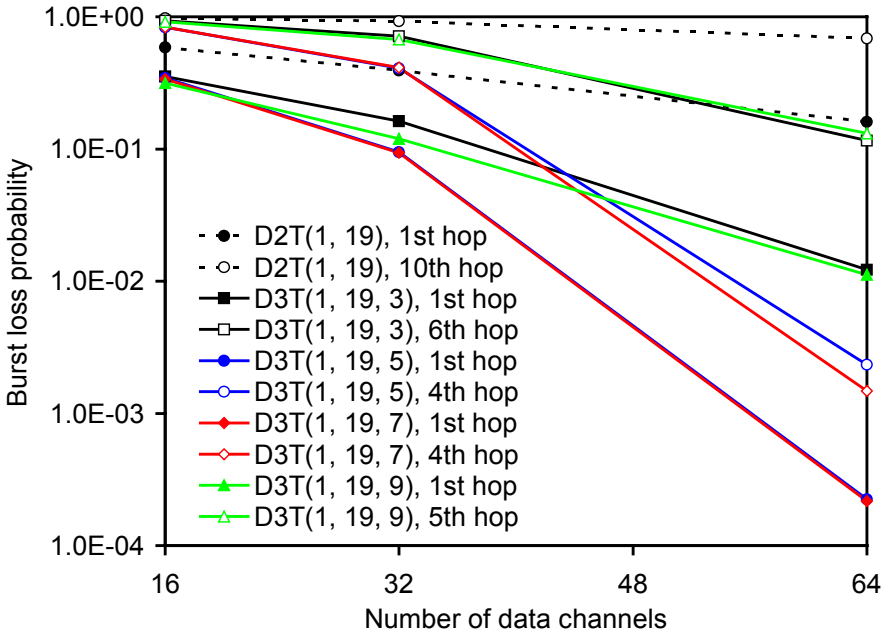


Fig. 4.2. Burst loss probability, as a function of the number of data channels per link (F), for D2T(1,19) and D3T(1,19,w_3) in the first and last hops, for JIT;

N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}=12.5μs.

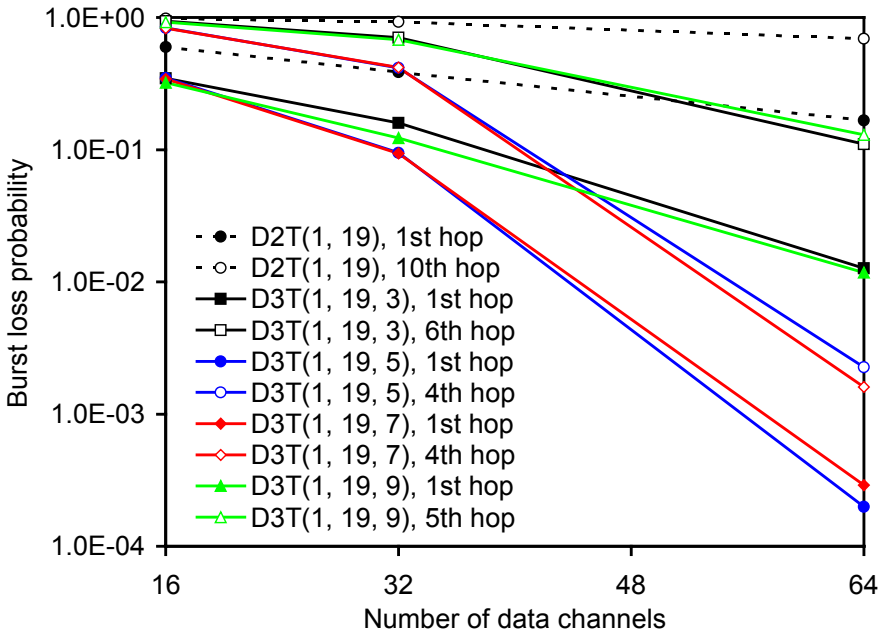


Fig. 4.3. Burst loss probability, as a function of the number of data channels per link (*F*), for D2T(1,19) and D3T(1,19,w_3) in the first and last hops, for JumpStart; *N*=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}=12.5μs.

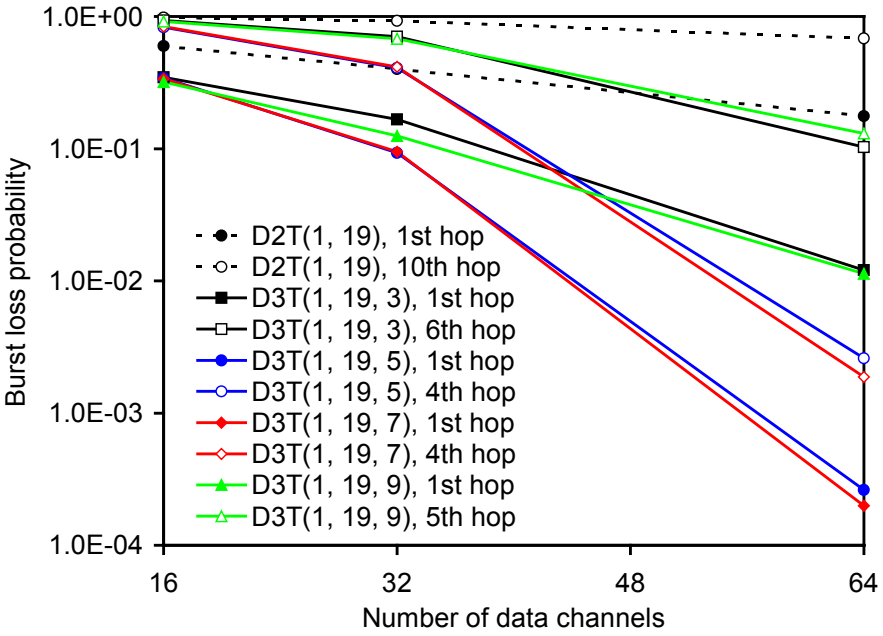


Fig. 4.4. Burst loss probability, as a function of the number of data channels per link (*F*), for D2T(1,19) and D3T(1,19,w_3) in the first and last hops, for JIT$^+$; *N*=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}=12.5μs.

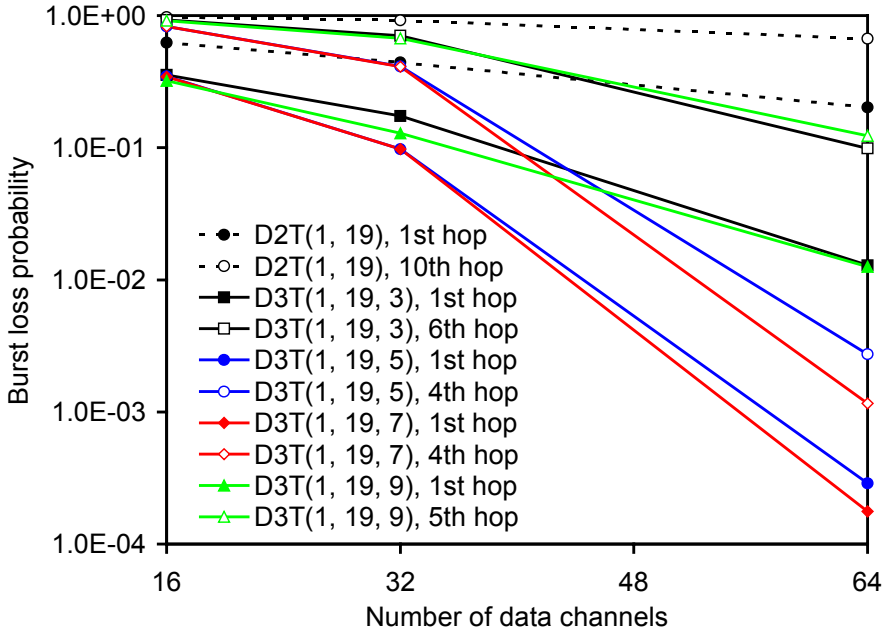


Fig. 4.5. Burst loss probability, as a function of the number of data channels per link (F), for D2T(1,19) and D3T(1,19,w_3) in the first and last hops, for JET;

N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}=50μs.

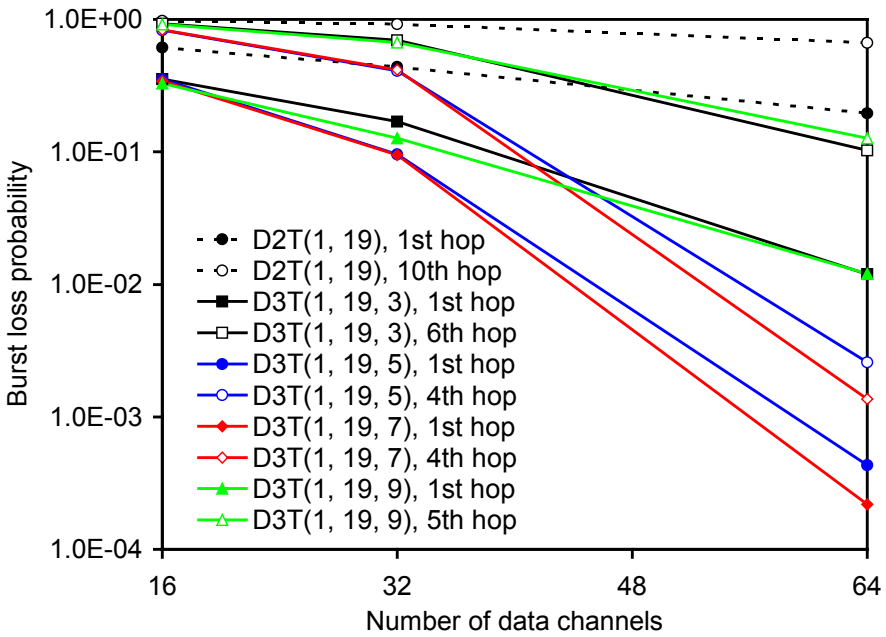


Fig. 4.6. Burst loss probability, as a function of the number of data channels per link (F), for D2T(1,19) and D3T(1,19,w_3) in the first and last hops, for Horizon;

N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}=25μs.

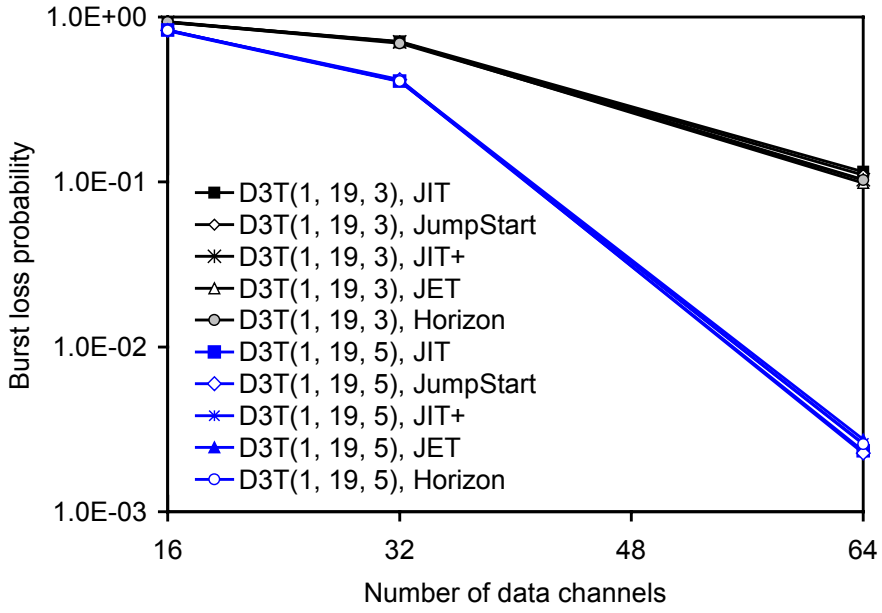


Fig. 4.7. Burst loss probability, as a function of the number of data channels per link (F), in the last hop of D3T(1,19,3) and D3T(1,19,5) for JIT, JumpStart, JIT$^+$, JET, and Horizon; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

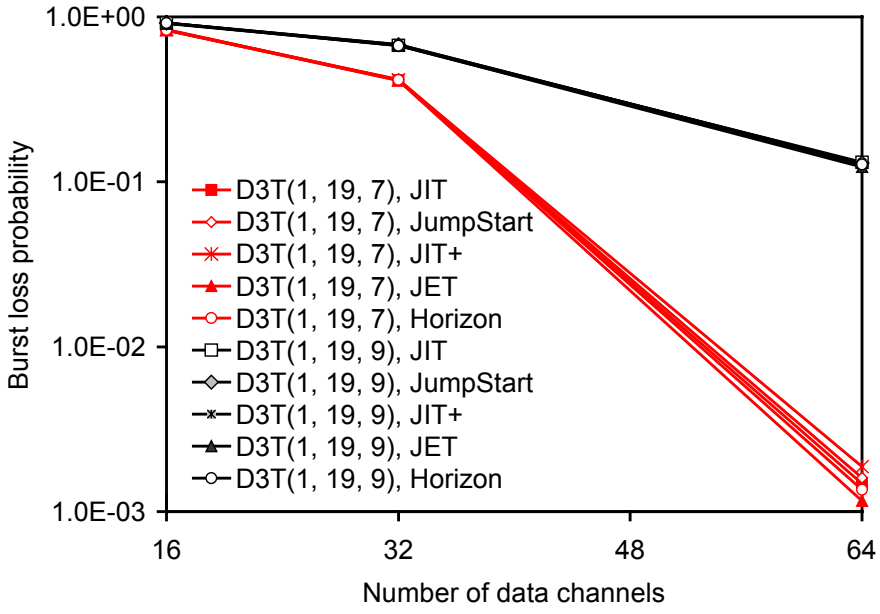


Fig. 4.8. Burst loss probability, as a function of the number of data channels per link (F), in the last hop D3T(1,19,7) and D3T(1,19,9) for JIT, JumpStart, JIT$^+$, JET, and Horizon; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

Figure 4.9 shows the burst loss probability for all hops of the D2T(1,19) and D3T(1,19,7) for JIT protocol. This figure clearly shows the performance improvement due to the increase of nodal degree from 2 (bi-directional ring topology) to 3 (chordal ring topology). For degree-three chordal ring, when a larger number of data channels are available (more than 64 data channels per link), the burst loss probability tends to zero. A similar performance has been observed for JumpStart, JIT$^+$, JET and Horizon. For this reason, the corresponding figures are not shown. This behavior is also confirmed in Figures 4.10 and 4.11, as a function of the number of hops, that show the burst loss probability for D2T(1,19), D3T(1,19,3), and D3T(1,19,5), and for D2T(1,19), D3T(1,19,7), and D3T(1,19,9), respectively, for JIT, JumpStart, JIT$^+$, JET and Horizon. These figures clearly show the better performance of the chordal rings with smallest network diameters, i.e. w_3=5 or w_3=7. These figures also show that chordal ring networks with chord lengths leading to larger network diameters (w_3=3 or w_3=9) may lead to higher burst loss probabilities.

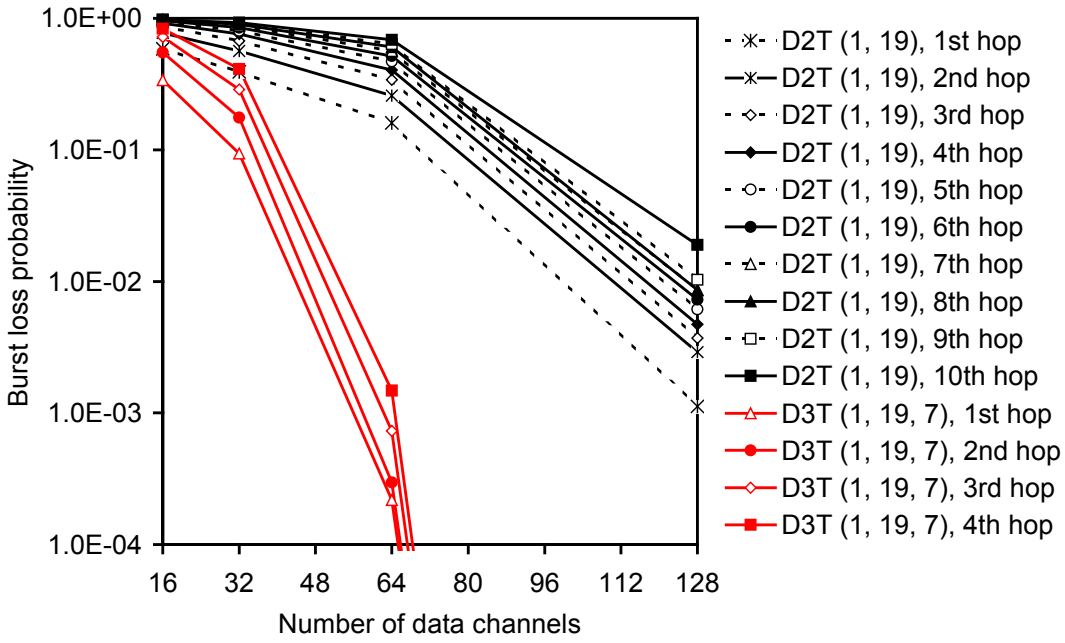


Fig. 4.9. Burst loss probability, as a function of the number of data channels per link (F), for D2T(1,19) and D2T(1,19,7) for JIT; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}=12.5μs.

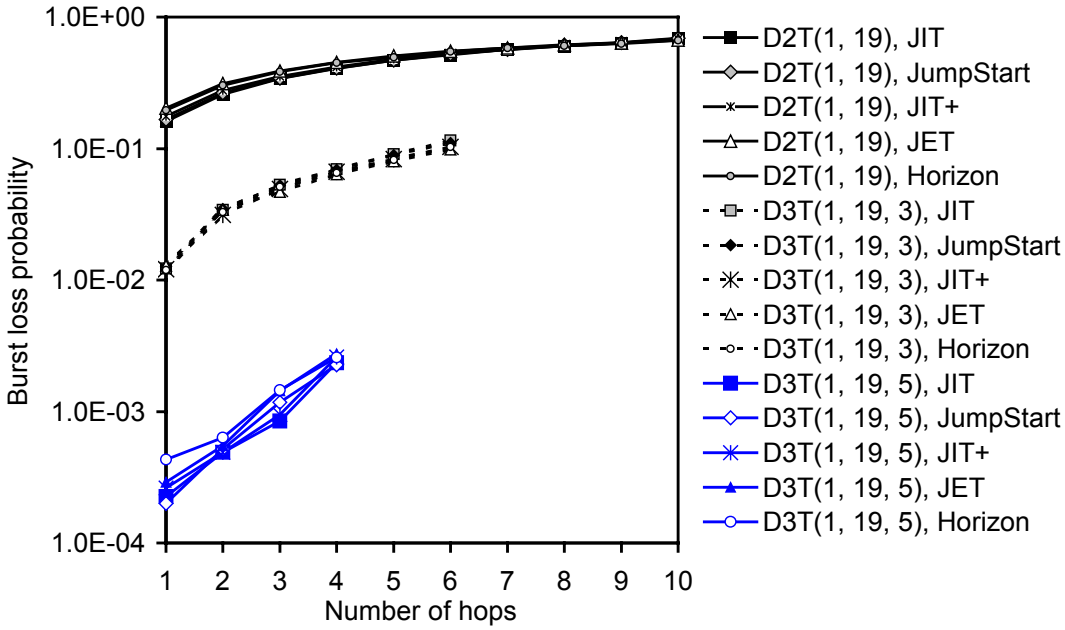


Fig. 4.10. Burst loss probability, as a function of the number of hopes, for D2T(1,19), D3T(1,19,3), and D3T(1,19,5) for JIT, JumpStart, JIT$^+$, JET, and Horizon, with F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

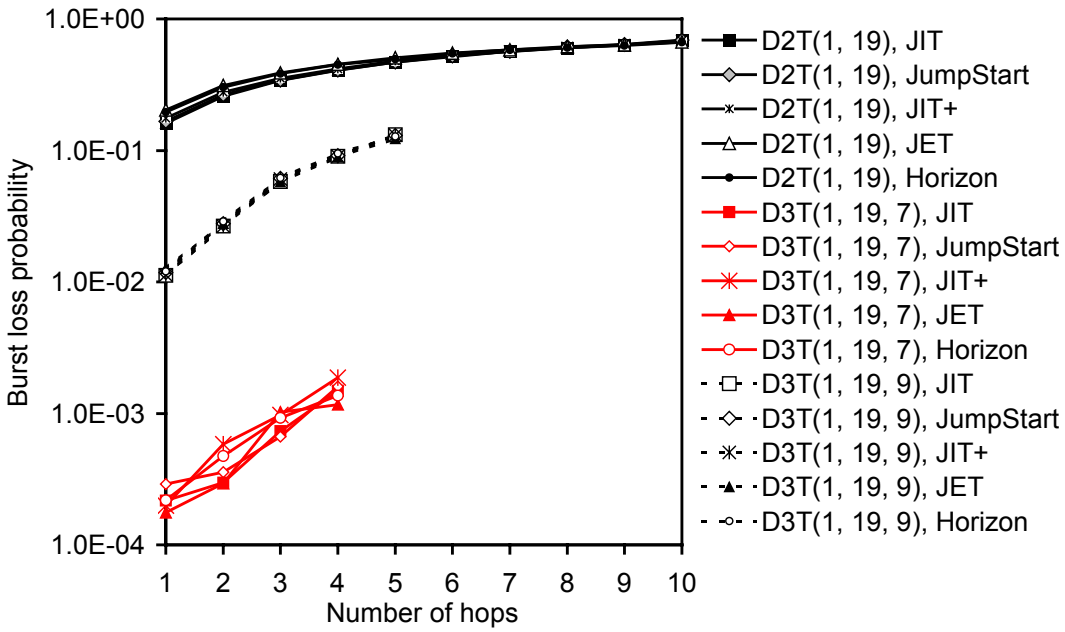


Fig. 4.11. Burst loss probability, as a function of the number of hopes, for D2T(1,19), D3T(1,19,7), and D3T(1,19,9) for JIT, JumpStart, JIT$^+$, JET, and Horizon, with F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

Figures 4.12 and 4.13 show the nodal degree gain, $G_{n,k}(i,j)$, for $1 \leq i,j \leq 4$ and $i=j$, due to the increase of nodal degree from 2 (D2T(1,19)) to 3 (D3T(1,19,5)) (Figure 4.12) and from 2 (D2T(1,19)) to 3 (D3T(1,19,7)) (Figure 4.13), for JIT, JumpStart, JIT$^+$, JET, and Horizon protocols. For F=16, or F=32, the nodal degree gain is very small due to the high burst loss probabilities observed for these numbers of data channels per link. However, when the number of data channels increases to 64, a nodal degree gain of about 3 orders of magnitude is observed in the first hop, which slightly decreases with the increase of the number of hops. To improve readability in Figures 4.12 and 4.13, the nodal degree gain for F=16 is not plotted because this gain is very small, as above-mentioned. In both figures is also possible to observe that the nodal degree gain decreases with the increasing of number of hops. This result confirms the results observed previously in Figures 4.2–4.6.

Figure 4.14 shows the increase of the nodal degree gain with the increase of the number of data channels per link, F, from 2 (D2T(1,19)) to 3 (D3T(1,19,7)) in the 4th hop of both topologies, $G_{2,3}(4,4)$, and in the last hop of both topologies $G_{2,3}(10,4)$. The D3T(1,19,7) topology was chosen because it leads to the best performance among degree-three topologies with 20 nodes. As may be seen, for F=64 data channels per link, the nodal degree gain for the last (10th) and the 4th hop are between two and three orders of magnitude, withthe gain obtained in the last hop being slightly better than in the 4th hop. When the number of data channels is small (F=16 and F=32) the nodal degree gain is less than one order of magnitude. Another important observation that can be made from this figure is that the network performance for the considered resource reservation protocols is very close.

Figure 4.15 shows the burst loss probability, as a function of the chord length (w_3), for the last hop of degree-three chordal ring networks with 20 nodes and F=64. This figure clearly shows that the best network performance is obtained for chordal rings with w_3=5 and w_3=7. This observation is confirmed in Figure 4.16, which shows the corresponding nodal degree gain. As may be seen, larger gains, between two and three orders of magnitude, are observed for w_3=5 and w_3=7, whereas a nodal degree gain smaller than 6 is observed for w_3=3 and w_3=9. As may also be seen, D3T(1,19,7) has slightly better performance than D3T(1,19,5).

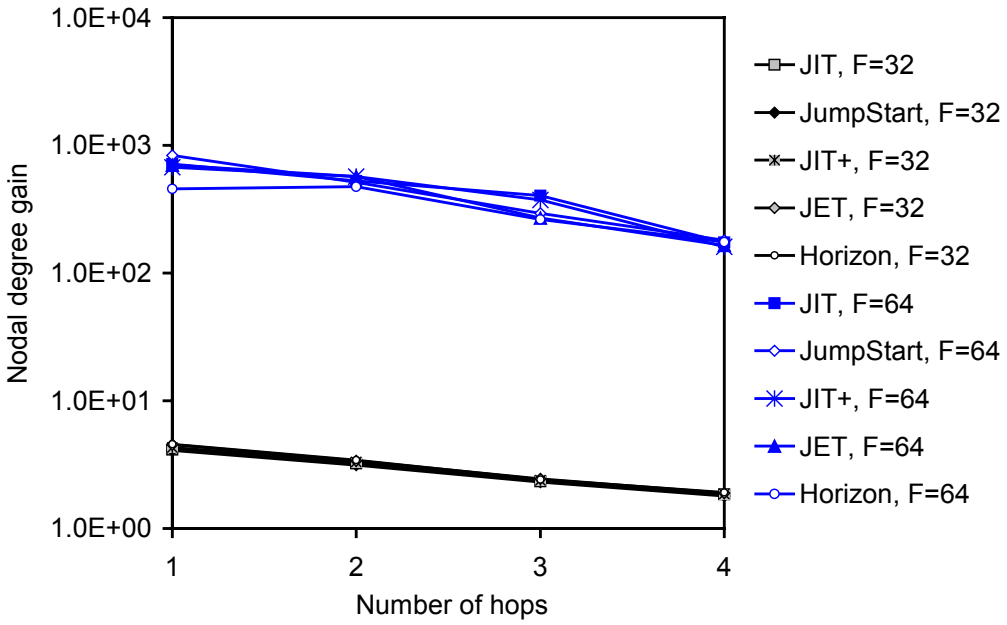


Fig. 4.12. Nodal degree gain, $G_{n,k}(i,j)$, for $1 \leq i, j \leq 4$ and $i \neq j$, due to the increase of nodal degree from 2 (D2T(1,19)) to 3 (D3T(1,19,5)) for JIT, JumpStart, JIT$^+$, JET, and Horizon; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs; F: number of data channels per link.

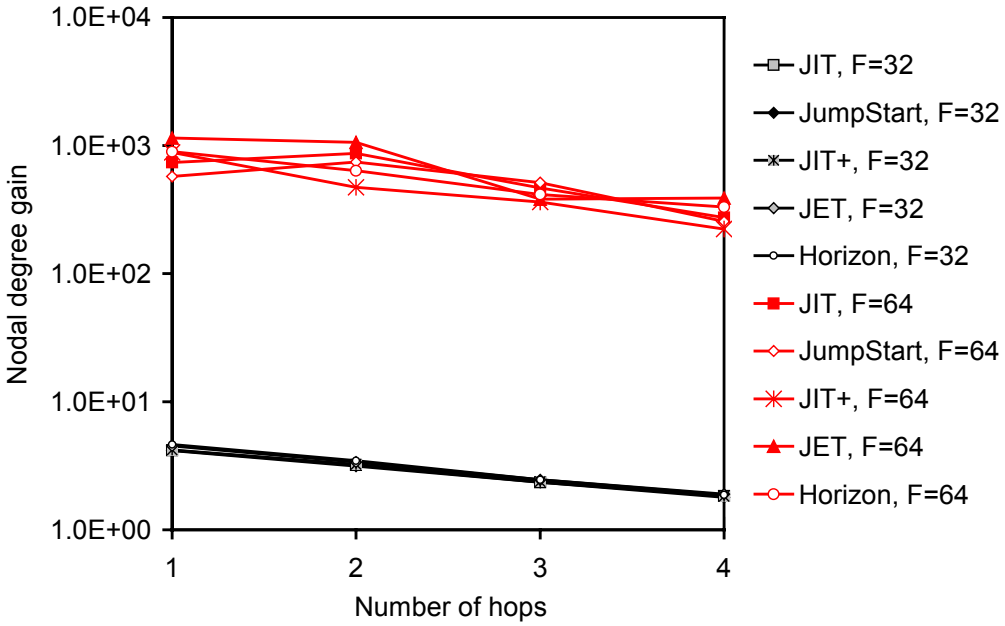


Fig. 4.13. Nodal degree gain, $G_{n,k}(i,j)$, for $1 \leq i, j \leq 4$ and $i \neq j$, due to the increase of nodal degree from 2 (D2T(1,19)) to 3 (D3T(1,19,7)) for JIT, JumpStart, JIT$^+$, JET, and Horizon; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs; F: number of data channels per link.

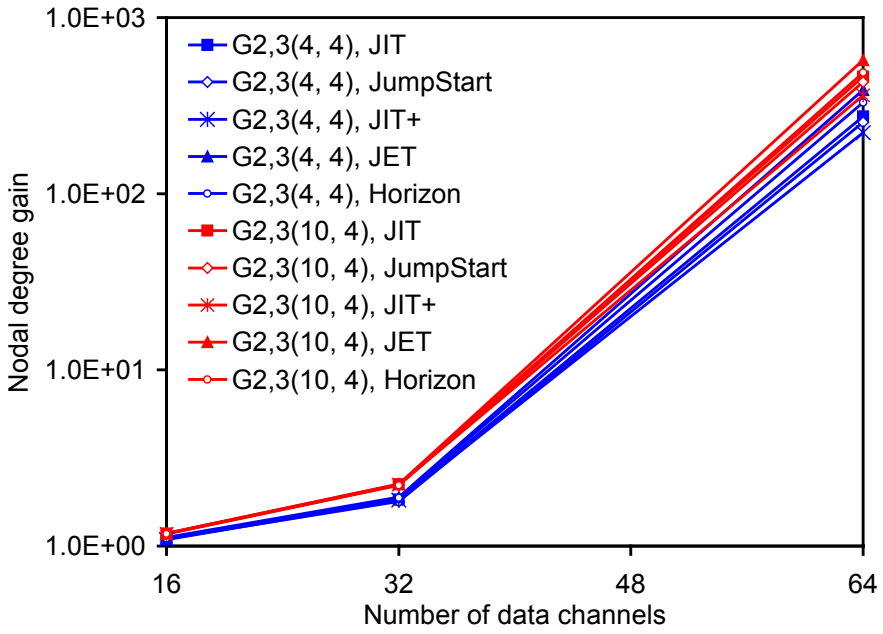


Fig. 4.14. Nodal degree gain due to the increase of nodal degree from 2 (D2T(1,19)) to 3 (D3T(1,19,7)) in the 4th hop of both topologies, i.e. $G_{2,3}(4,4)$, and in the last hop of both topologies, i.e. $G_{2,3}(10,4)$; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

Figure 4.17 illustrates the burst loss probability, as a function of λ/μ, in the last hop of D3T(1,19,5) and D3T(1,19,7), for JIT, JumpStart, JIT$^+$, JET and Horizon, with F=64 data channels per link. As above-mentioned, D3T(1,19,5) and D3T(1,19,7) are the degree-three chordal ring topologies with best performance. As may be seen, the burst loss probability increases with the increase of λ/μ. The behavior of both topologies is similar for each resource reservation protocols, as may be seen in Figure 4.17. For values of λ/μ less than 25.6 the result of burst loss probability tends to zero. Figure 4.18 shows the corresponding nodal degree gain as a function of λ/μ for the last hop of both topologies (D3T(1,19,5) and D3T(1,19,7)). As may be seen for both topologies, the largest gains, around four orders of magnitude, are observed for the smallest values of λ/μ (for λ/μ=25.6). For larger values of λ/μ (greater or equal than 51.2) the nodal degree gain is less than one order of magnitude. These results also confirm previous observations in terms of the performance comparison of the five resource reservation protocols considered in this study, i.e. their performance is very close.

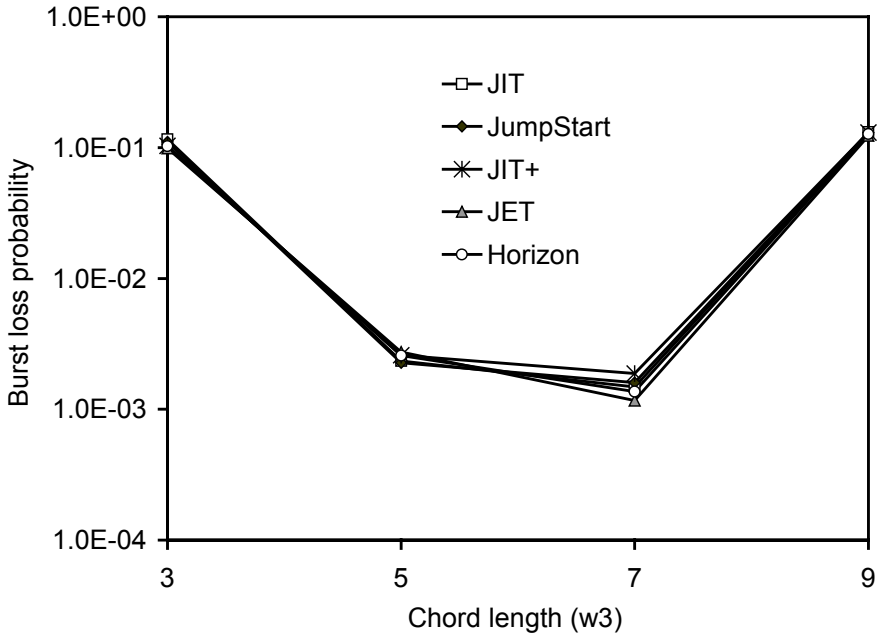


Fig. 4.15. Burst loss probability as a function of the chord length in the last hop of each D3T(1,19,w3) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

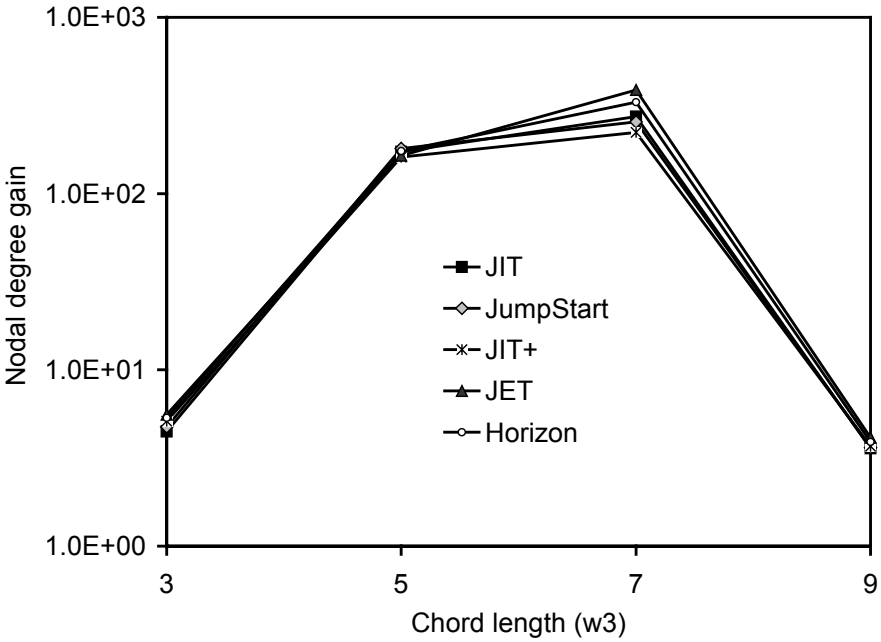


Fig. 4.16. Nodal degree gain, as a function of Chord length (w_3) for last hop of each D3T(1,19,w3) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

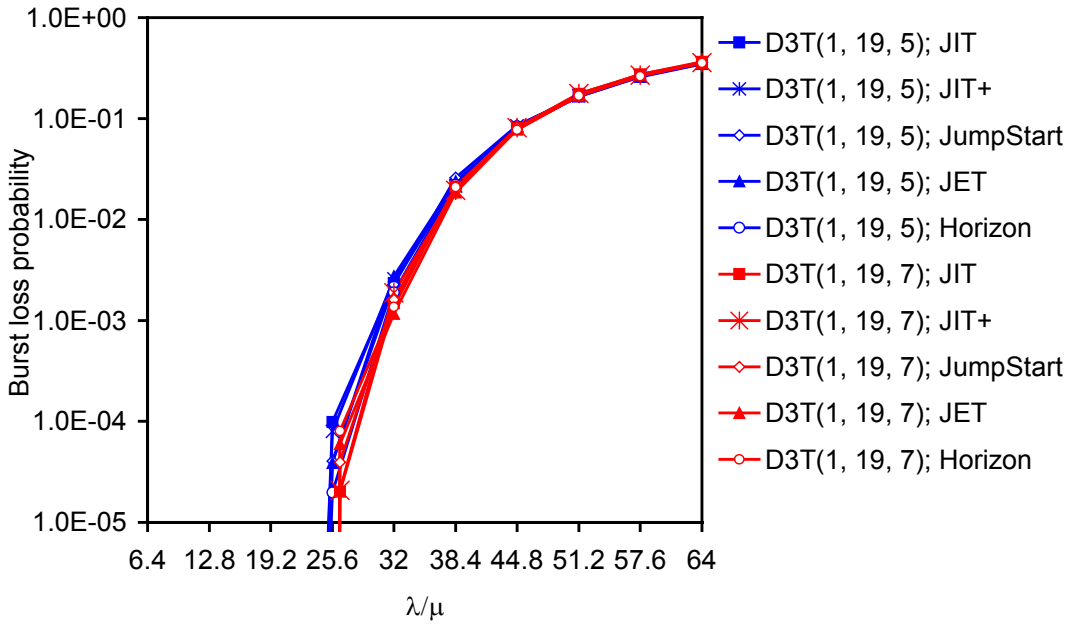


Fig. 4.17. Burst loss probability as a function of λ/μ for last hop of D3T(1,19,5) and D3T(1,19,7), for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

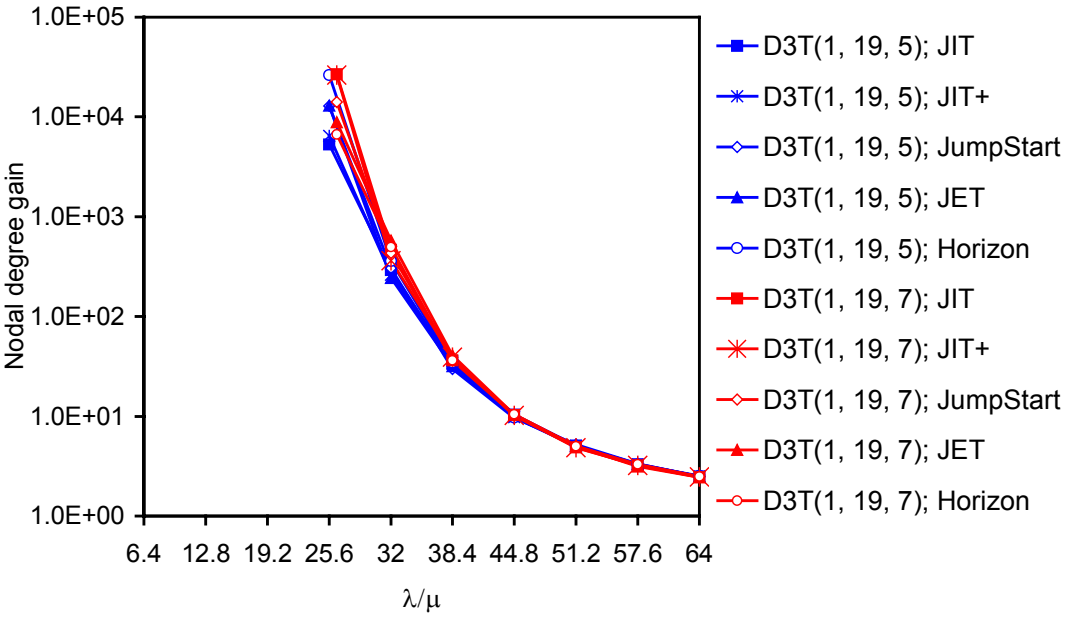


Fig. 4.18. Nodal degree gain, as a function of λ/μ for last hop of D3T(1,19,5) and D3T(1,19,7), for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

4.3.2 Influence of Chord Length Gain

This sub-section focuses on the influence of the chord length in the performance of OBS ring and degree-three chordal ring networks.

Figure 4.19 shows the chord length gain, G_{cl} (6,4; 3,w_3^*), as a function of the number of data channels, in the last hop of each D3T(1,19,w_3^*) for w_3^*=5 and w_3^*=7. As can be seen, the chord length gain in the last hop of each topology is very small for F=16 or F=32 data channels per link, due to the high burst loss probabilities observed for these number of data channels. It is possible to verify that the chord length gain in these cases is similar. For F=64 data channels, the largest chord length gain in the last hop is obtained for w_3^*=7, which is slightly less than two orders of magnitude.

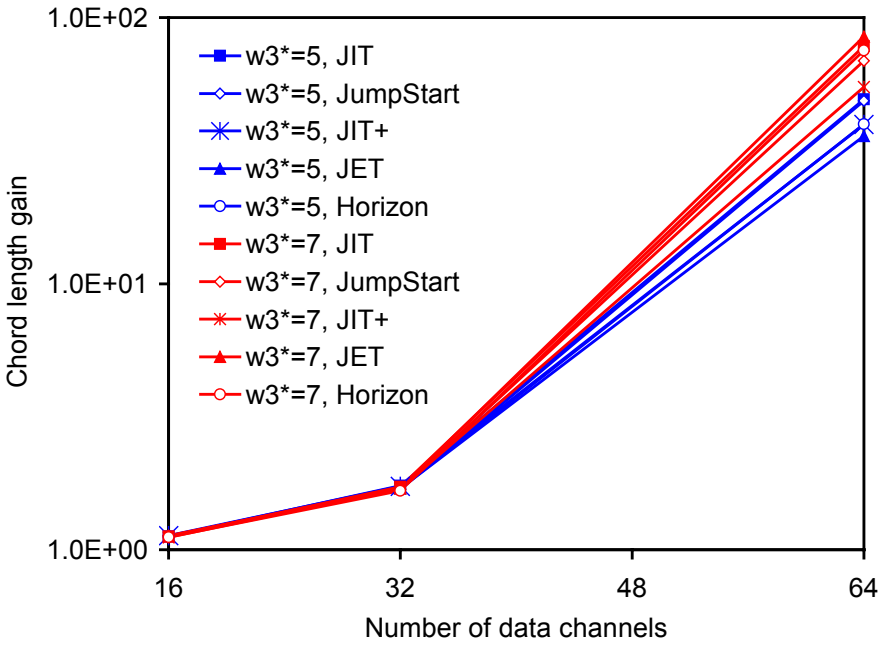


Fig. 4.19. Chord length gain G_{cl} (6,4; 3,w_3^*), as a function of number of data channels, in the last hop of D3T(1,19,w_3) for the choice of w_3^*=5 or w_3^*=7, instead of w_3=3; F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

Figure 4.20 shows the chord length gains G_{cl}(6,j; 3,w_3^*) in the last hop of each D3T(1,19,w_3), as a function of the chord length for JIT, JumpStart, JIT$^+$, JET, and

Horizon. This figure clearly shows that chord lengths of w_3=5 and w_3=7 lead to the best performances, being the performance for w_3=7 slightly better than the performance for w_3=5. The gain obtained for these chord lengths is between one and two orders of magnitude, which confirms previous observations.

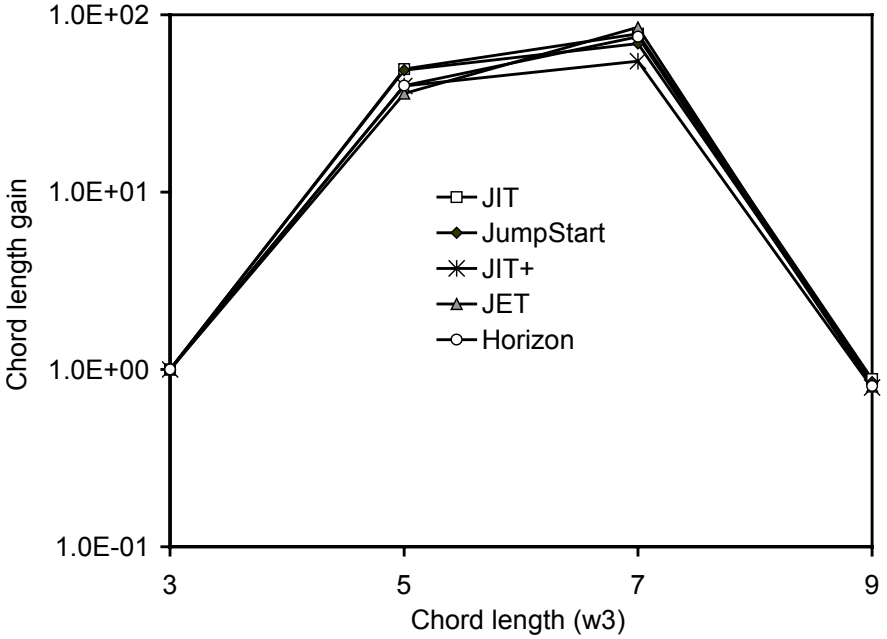


Fig. 4.20. Chord length gain $G_{cl}(6,j;\ 3,w_3^*)$ as a function of the chord length, in the last hop of each D3T(1,19,w_3^*) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

4.3.3 Effects of Setup Message Processing Time and OXC Configuration Time

This sub-section presents a study of the effect of the setup message processing time and the optical cross-connect (OXC) configuration time in the performance of OBS networks with ring and degree-three chordal ring topologies for JIT, JumpStart, JIT$^+$, JET, and Horizon protocols.

Figure 4.21 plots the burst loss probability as a function of OXC configuration time in the last hop of D2T(1,19) and D3T(1,19,7) for the five protocols under study, with F=64 and λ/μ=32. In this figure, a fixed value for T_{Setup} time is assumed for JIT, JumpStart, and JIT$^+$ as the value defined for JITPAC controllers [80, 141] and it is

estimated for JET and Horizon for current available technology. T_{OXC} is assumed to range from the value estimated for the near future scenario (T_{OXC}=20μs) up to ten times the value defined for current available technology, i.e. T_{OXC}=10*10ms=100ms. As may be seen in this figure, chordal rings clearly have better performance than rings for T_{OXC}≤50ms. It may also be observed that, for T_{OXC}≤1ms, the performance of chordal rings is independent of the change of the T_{OXC}, which means that a reduction of the values of T_{OXC} below 1ms does not improve the network performance. Moreover, it may also be observed that the relative performance of the five resource reservation protocols is similar, being JIT and JIT$^+$ slightly better than the other ones. For rings, the burst loss probability is very high and the increase of the T_{OXC} only slightly decreases the network performance. For values of T_{OXC} below 1ms it does not improve the network performance, such as in chordal rings. The performance of the five protocols in rings is very close.

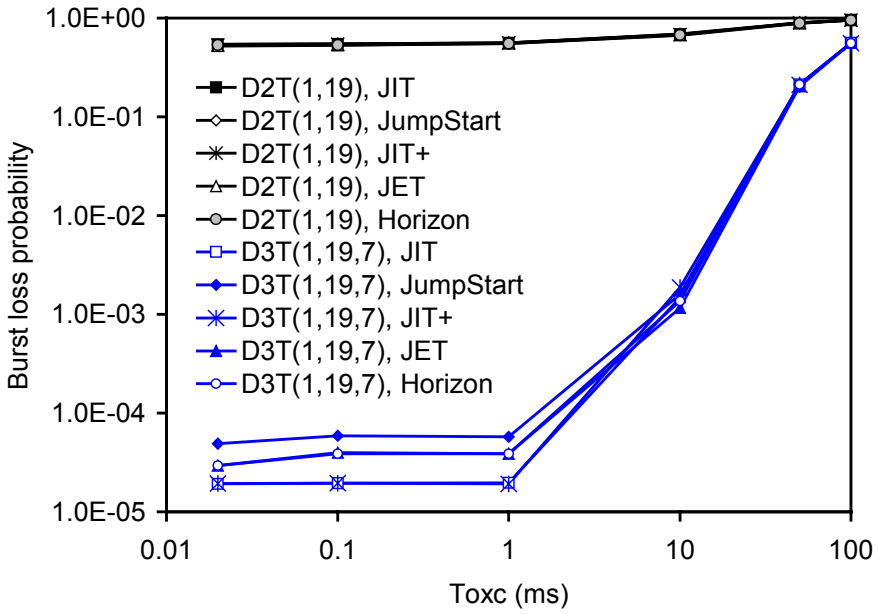


Fig. 4.21. Burst loss probability as a function of OXC configuration time in the last hop of D2T(1,19) and D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; λ/μ=32; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)= 50μs; T_{Setup}(Horizon)=25μs.

Figure 4.22 illustrates the burst loss probability as a function of *setup message* processing time (T_{Setup}) in the last hop of D2T(1,19) and D3T(1,19,7) for the five protocols under study, with F=64 and λ/μ=32. Two scenarios are considered regarding T_{OXC}: it assumes the value for a current available technology (T_{OXC}=10ms)

and an estimated value for a near future scenario (T_{OXC}=20μs). For each curve of Figure 4.22, T_{OXC} is assumed to have a fix value while T_{Setup} ranges between the values considered for the current available technology and the estimated values for the near future technology. Thus, T_{Setup} ranges between 12.5μs and 1μs, for JIT, JumpStart, and JIT$^+$, ranges between 25μs and 2μs for JET, and between 50μs and 4μs for Horizon. As may be seen in this figure, the performance of chordal rings is clearly better than rings and the behavior of the five protocols is very close. As may be observed, the reduction of T_{Setup} does not lead to a better network performance. It may also be observed that for chordal rings a reduction of the T_{OXC} from 10ms down to 20μs leads to a performance improvement, of about two orders of magnitude, which confirms the results presented in Figure 4.21. For rings, the burst loss is so high that both the reduction of T_{OXC} and different values of T_{Setup} do not have impact on the network performance.

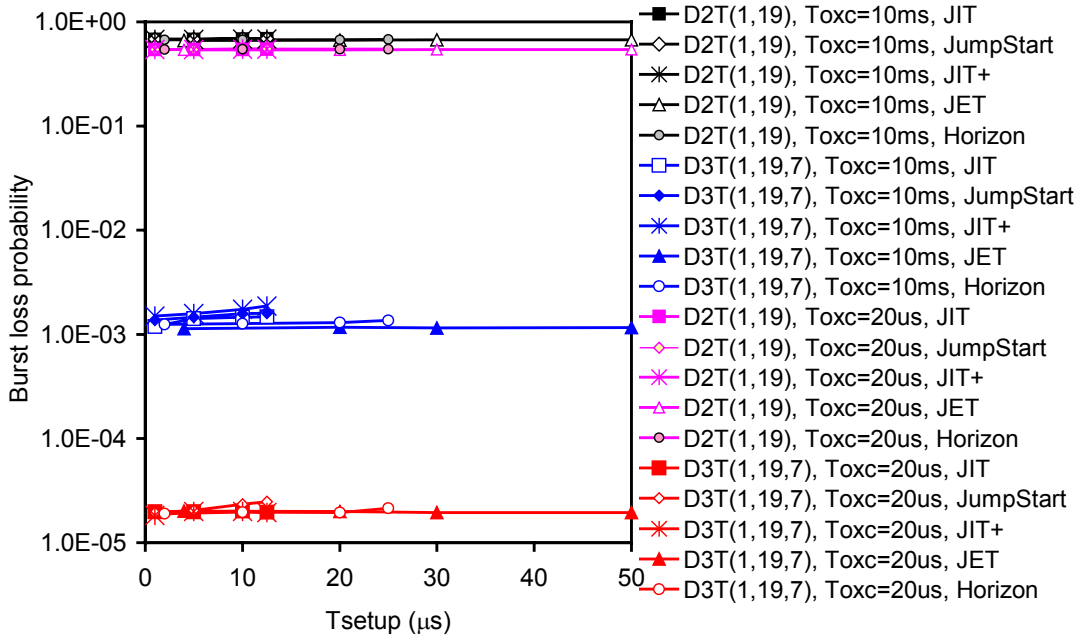


Fig. 4.22. Burst loss probability as a function of setup message processing time in the last hop of D2T(1,19) and D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; λ/μ=32; T_{OXC}=10ms; T_{OXC}=20μs.

After studying OBS ring and chordal ring networks with fixed T_{OXC} and fixed T_{Setup}, respectively, in Figure 4.23 it is assumed that the change of T_{Setup} is a function of the variation of T_{OXC} according to a linear interpolation. It is assumed that T_{OXC} ranges between 20μs and 100ms, considering the following intermediate

times: 0.1ms, 1ms, 10ms, and 50ms. Therefore, the value of T_{Setup} for JIT, JumpStart, and JIT$^+$ protocols, where X is the correspondent resource reservation protocol is given by:

$$T_{Setup}(X) = 1 + \frac{11.5}{10^4 - 20}(T_{OXC}(X) - 20) \quad (\mu s) \tag{4.3}$$

T_{Setup} for JET protocol is given by:

$$T_{Setup}(JET) = 4 + \frac{46}{10^4 - 20}(T_{OXC}(JET) - 20) \quad (\mu s) \tag{4.4}$$

For Horizon resource reservation protocol T_{Setup} is given by:

$$T_{Setup}(Horizon) = 2 + \frac{23}{10^4 - 20}(T_{OXC}(Horizon) - 20) \quad (\mu s) \tag{4.5}$$

Figure 4.23 shows the burst loss probability as a function of OXC configuration time in the last hop of D2T(1,19) and D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon, being T_{Setup} computed according to (4.3), (4.4), and (4.5) for each protocol. As may be seen, for T_{OXC}<0.1ms, a small change in the burst loss can be observed in Figure 4.21. However, this small change is not significant in terms of network performance and therefore, this figure also confirms the previous observations about the influence of T_{Setup} and T_{OXC} in the network performance (Figures 4.21 and 4.22).

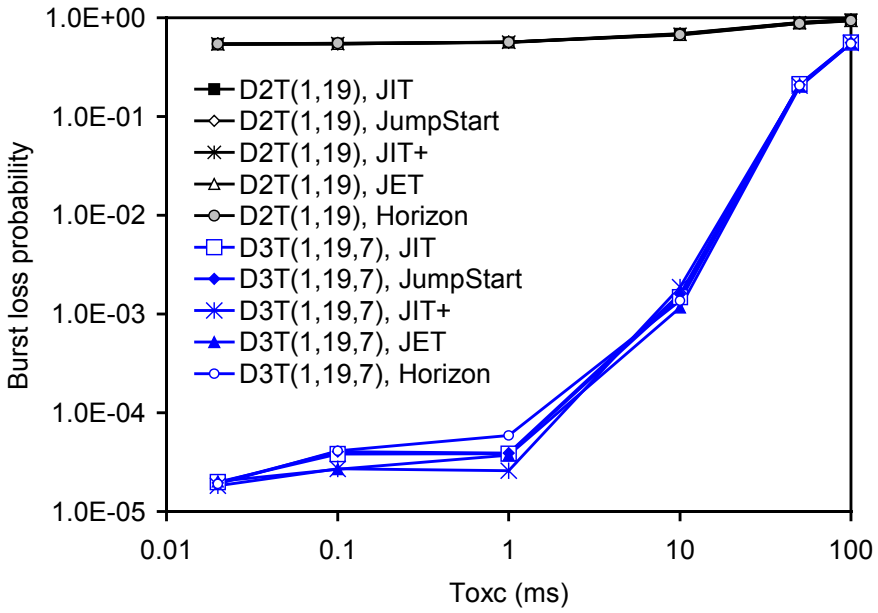


Fig. 4.23. Burst loss probability as a function of OXC configuration time in the last hop of D2T(1,19) and D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; N=20; F=64; λ/μ=32;with changing T_{Setup} according to (4.3), (4.4), and (4.5) for each resource reservation protocol.

Figure 4.24 assumes that the change of T_{OXC} is a function of the variation of T_{Setup} and it is obtained by solving equations (4.3), (4.4), and (4.5) regarding T_{OXC}. T_{Setup} assumes the value ranging between 1μs and 12.5μs, considering 5μs and 10μs as intermediate times, for JIT, JumpStart, and JIT$^+$ protocols. Therefore, the value of T_{OXC} for these protocols, where X is the correspondent resource reservation protocol is given by:

$$T_{OXC}(x) = 20 + \frac{(T_{Setup}(x) - 1)(10^4 - 20)}{11.5} \quad (\mu s) \tag{4.6}$$

T_{Setup} for JET assumes the value ranging between 4μs and 50μs, considering 20μs and 30μs as intermediate times. T_{OXC} for JET is given by:

$$T_{OXC}(JET) = 20 + \frac{(T_{Setup}(JET) - 4)(10^4 - 20)}{46} \quad (\mu s) \tag{4.7}$$

T_{Setup} for Horizon assumes the value ranging between 2μs and 25μs, considering 10μs and 20μs as intermediate times. T_{OXC} for Horizon is given by:

$$T_{OXC}(Horizon) = 20 + \frac{(T_{Setup}(Horizon) - 2)(10^4 - 20)}{23} \quad (\mu s) \tag{4.8}$$

Figure 4.24 shows the burst loss probability as a function of Setup message processing time in the last hop of D2T(1,19) and D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon, being T_{OXC} computed according to (4.6), (4.7), and (4.8) for each protocol. As may be seen, for ring network, the performance presents a small change that is not significant. However, for degree-three chordal ring, a similar performance is observed for the values considered for T_{OXC} in function of T_{Setup}. When the value of T_{Setup} increases, the correspondent burst loss probability also increases. For values ranging between the first and the last times, considered for each protocol T_{Setup}, the burst loss probability increases around two orders of magnitude. This figure also confirms previous results in terms of the best performance of chordal rings in comparison with rings and the influence of T_{OXC} and T_{Setup} in the network performance.

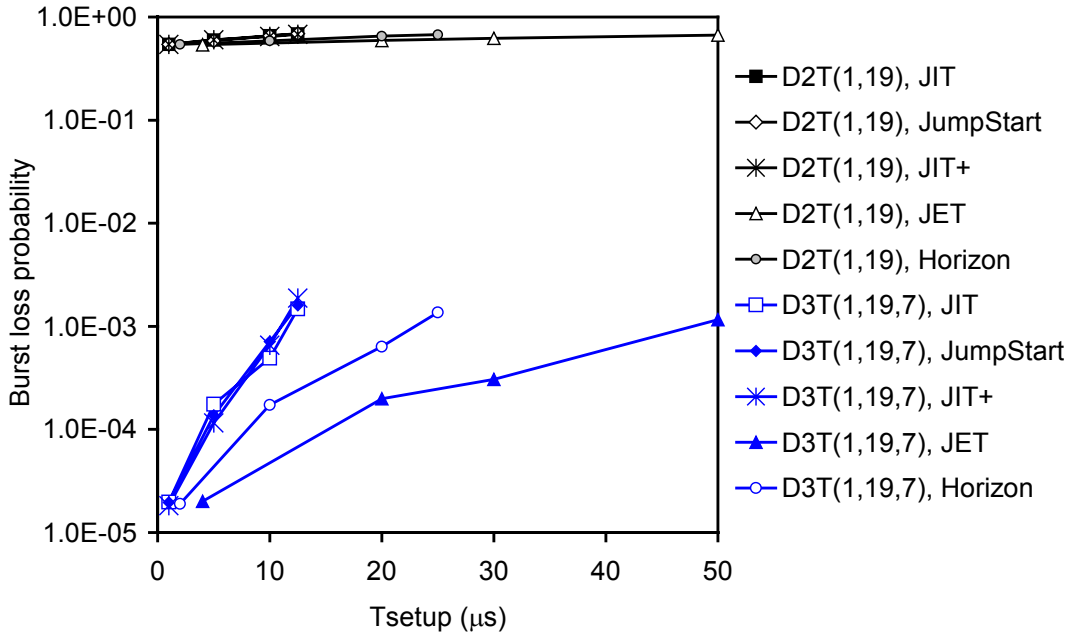


Fig. 4.24. Burst loss probability *versus* T_{Setup} in the last hop of D2T(1,19) and D3T(1,19,7) for JIT, JumpStart, JIT+, JET, and Horizon; F=64; N=20; λ/μ=32; with changing T_{OXC} according to (4.6), (4.7), and (4.8) for each resource reservation protocol.

Next figures focus on degree-three chordal ring topologies. Figures 4.25 and 4.26 show the burst loss probability as a function of OXC configuration time in the last hop of D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon protocols, with λ/μ=32, considering F=32 and F=64 data channels per link. In Figure 4.25 a fixed value for the setup message processing time (T_{Setup}) is assumed, which is the value defined for JIT, JumpStart, and JIT$^+$ (T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs) and estimated for JET and Horizon (T_{Setup}(JET)=50μs and T_{Setup}(Horizon)=25μs) for current available technology. On the other hand, in Figure 4.26 it is assumed that the change of T_{Setup} is a function of the OXC configuration time (T_{OXC}) given by equations 4.3, 4.4, and 4.5 above-mentioned. The influence of T_{Setup} variation, as a function of T_{OXC}, is very small. As may be seen, the performance of this network is clearly better when a large number of data channels are available (F=64). When the value of the T_{OXC} increases, the difference between network performances of both number of data channels decreases. When the value of T_{OXC} is small (T_{OXC} ≤1ms) the performance of the five protocols remains more or less with a constant value. This means that even if the technology is improved, this advance will not produce better behavior in terms of network performance. It is observed that the performance of the five resource

reservation protocols is very close for F=32 and for F=64 with T_{OXC}>1ms. The behavior of this network is the same both with fixed T_{Setup} (Figure 4.25) and with a changed T_{Setup} in function of T_{OXC} (Figure 4.26). A figure considering the burst loss probability as a function of setup message processing time, in the last hop of D3T(1,19,7) for the same resource reservation protocols with F=32 and F=64 is not shown because a similar result was obtained. Values of T_{OXC} were computed according to (4.6), (4.7), and (4.8) for each protocol.

Figures 4.27, 4.28, and 4.29 also confirm previous observations. They show the burst loss probability as a function of OXC configuration time in the last hop of D3T(1,19,7) with λ/μ=32 and λ/μ=44.8 for JIT, JumpStart, JIT$^+$, JET, and Horizon with F=64 data channels per link. As shown in two previous figures, a fixed value of T_{Setup} (Figure 4.27), a changed T_{Setup} value in function of T_{OXC} (Figure 4.28), and a changed T_{OXC} value in function of T_{Setup} are assumed (Figure 4.29). As may be seen, the performance of this network topology with λ/μ=32 in both figures is similar to its performance in both previous figures with F=64. Concerning the performance comparison with λ/μ=44.8 in Figures 4.27 and 4.28, it is very similar. In terms of other conclusions about these figures, they coincide with those present for Figures 4.25 and 4.26.

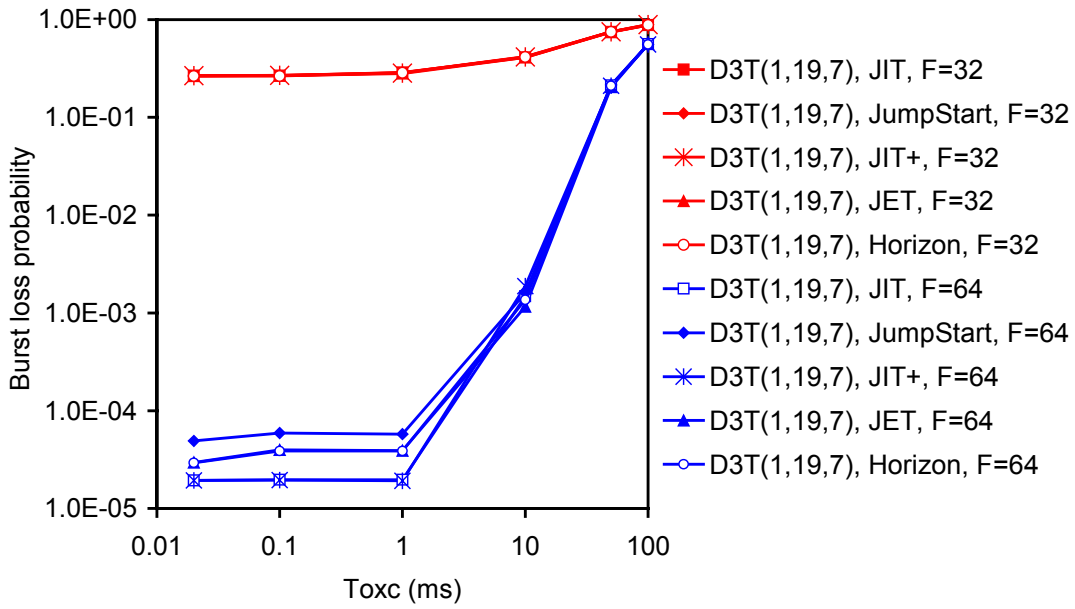


Fig. 4.25. Burst loss probability as a function of OXC configuration time in the last hop of D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=32 and F=64; N=20; λ/μ=32; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

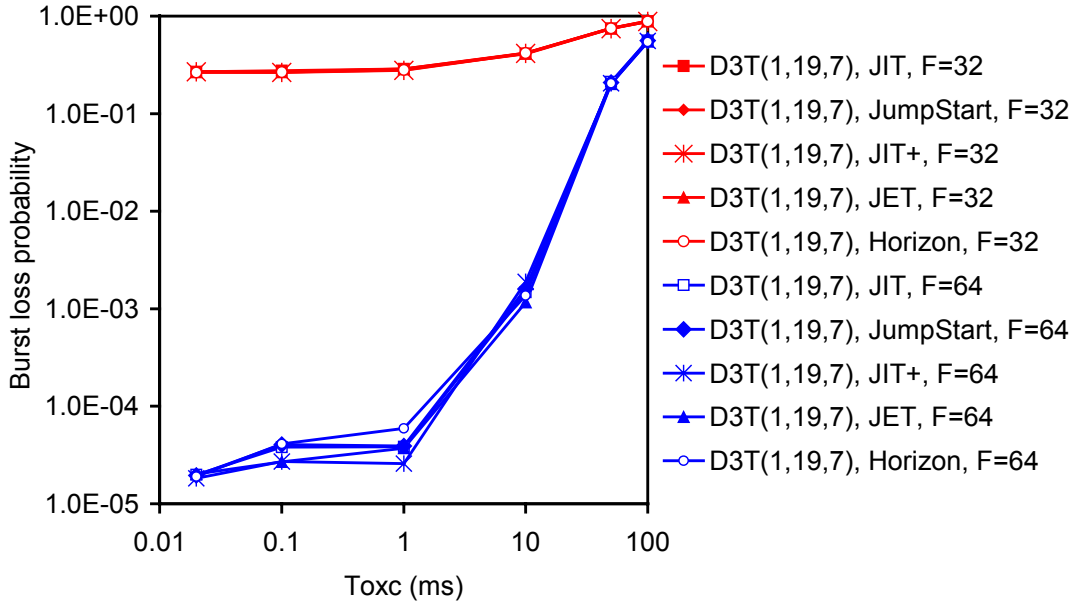


Fig. 4.26. Burst loss probability as a function of OXC configuration time in the last hop of D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=32 and F=64; N=20; λ/μ=32; with changing T_{Setup} according to (4.3), (4.4), and (4.5) for each resource reservation protocol.

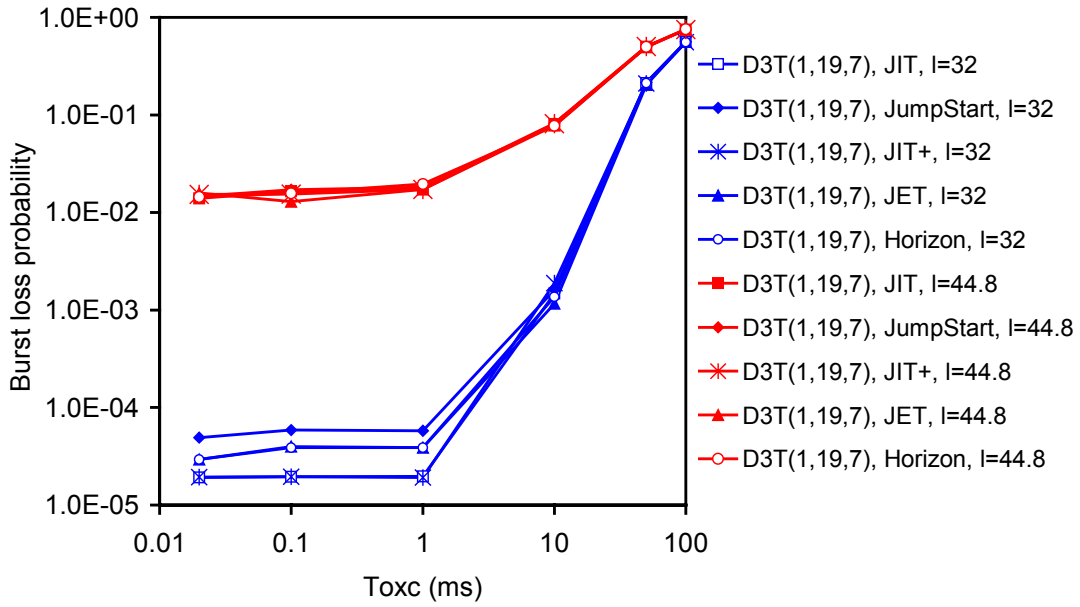


Fig. 4.27. Burst loss probability as a function of OXC configuration time in the last hop of D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; $l\equiv\lambda/\mu$=32 and $l\equiv\lambda/\mu$=44.8; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

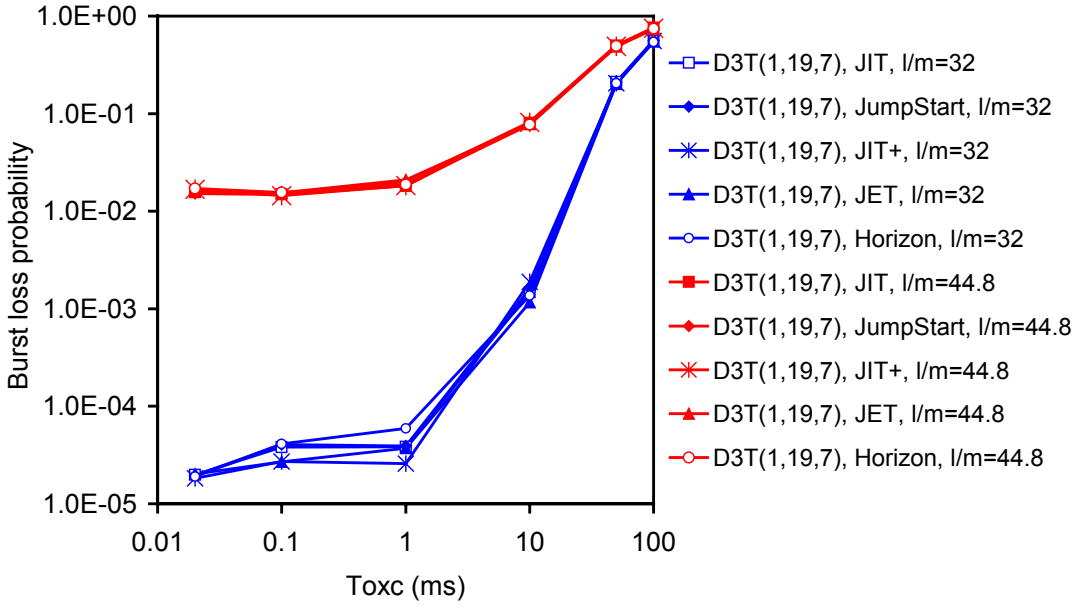


Fig. 4.28. Burst loss probability as a function of OXC configuration time in the last hop of D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; l/m≡λ/μ=32 and l/m≡λ/μ=44.8; with changing T_{Setup} according to (4.3), (4.4), and (4.5) for each resource reservation protocol.

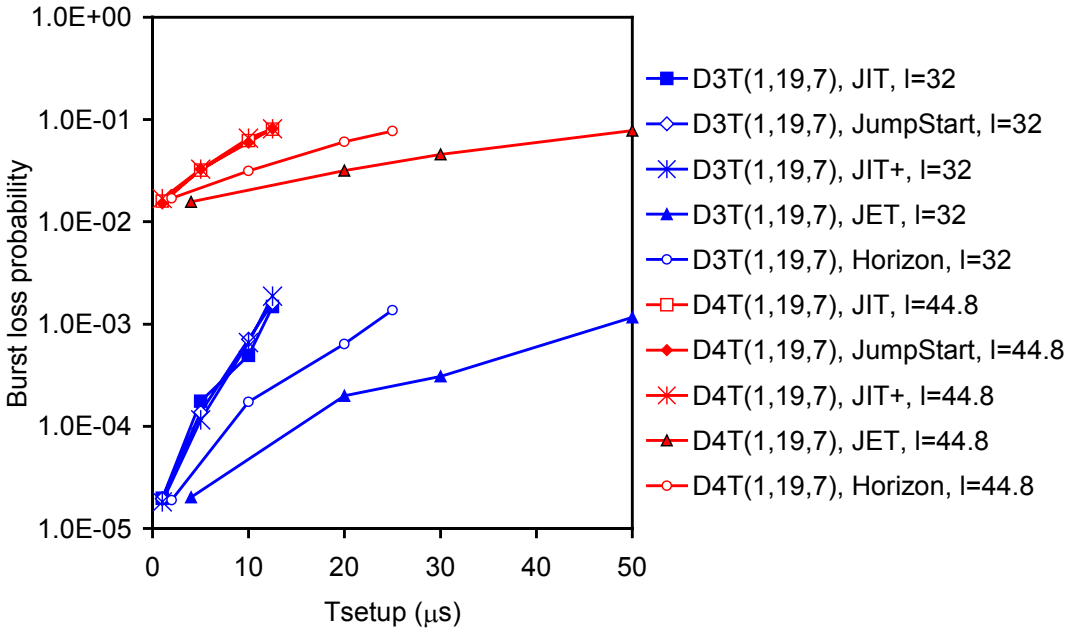


Fig. 4.29. Burst loss probability as a function of Setup message processing time in the last hop of D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; l≡λ/μ=32 and l≡λ/μ=44.8; with changing T_{OXC} according to (4.6), (4.7), and (4.8) for each resource reservation protocol.

4.4 Performance Assessment of OBS Degree-four Chordal Ring Networks

This section will pay attention on the loss of data bursts in OBS networks, introducing degree-four chordal ring topologies in comparison with chordal ring topologies with degree three. For comparison purposes, ring topologies are also considered.

4.4.1 Impact of Chord Length

The focus of this sub-section is on the impact of the chord length in the performance of OBS networks with degree-four chordal ring topologies. Rings and, mainly, degree-three chordal ring topologies are also considered for comparison purposes.

Figure 4.30 shows the network diameter as a function of chord length for D3T(1,19,w_3), D4T(1,19,3,w_4), and D4T(1,19,5,w_4). As may be seen, for N=20 nodes, the maximum and minimum diameter of the degree-three chordal ring family are 6 and 4, respectively, while, the maximum and minimum diameter of the degree-four chordal ring family are 4 and 3, respectively. For degree-three topology families with 20 nodes (D3T(1,19,w_3)), the network diameter for chord length equal to 11, 13, 15, and 17 is symmetric regarding the chord length equal to 3, 5, 7, and 9. For D4T(1,19,3,w_4) topology, the network diameter for chord length equal to 11, 13, 15, and 17 is symmetric regarding the chord length equal to 5, 7, 9, and 11. For D4T(1,19,5,w_4) topology, the network diameter for chord length equal to 11, 13, 15, and 17 is symmetric regarding the chord length equal to 3, 7, and 9. Therefore, in Figure 4.31, for degree-three topology are considered chords length (w_3) equal to 3, 5, 7, and 9, for D4T(1,19,3,w_4) are considered chords length (w_4) equal to 5, 7, 9, and 11, and for D4T(1,19,5,w_4) are considered chords length (w_4) equal to 3, 7, and 9.

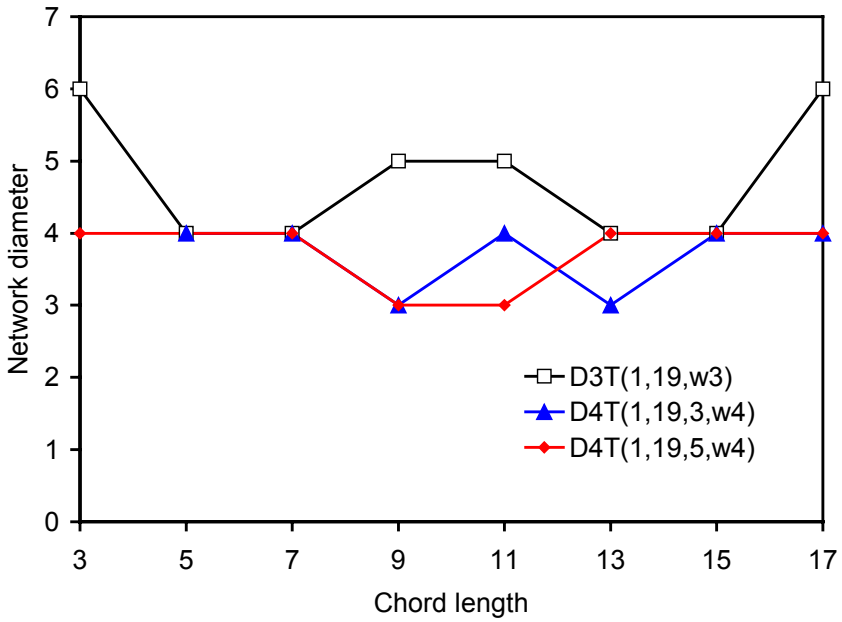


Fig. 4.30. Network diameter, as a function of chord length for D3T(1,19,w_3), D4T(1,19,3,w_4), and D4T(1,19,5,w_4).

Figure 4.31 shows the burst loss probability, with N=20 nodes, in function of chord length for degree-three (D3T(1,19,w_3)) and degree-four topologies (D4T(1,19,3,w_4) and (D4T(1,19,5,w_4)), for F=64 data channels per link. As may be seen, for D3T(1,19,w_3), the best network performance is obtained for w_3=5 and w_3=7, which corresponds to the minimum network diameter of 4 for this topology (see Figure 4.30). For D4T(1,19,3,w_4), the best network performance is obtained for w_4=9 and w_4=13, which correspond to the minimum network diameter of 3 for this topology. For D4T(1,19,5,w_4), the best network performance is obtained for w_4=9 and w_4=11, which correspond to the minimum network diameter of 3 for this topology. As may be seen, for the degree-four topologies, the chord length w_4=9 has slightly better performance for both topologies. Therefore, D4T(1,19,3,9) and (D4T(1,19,5,9) are chosen as degree-four chordal ring network topologies with the best performance, with slightly better behavior for D4T(1,19,3,9).

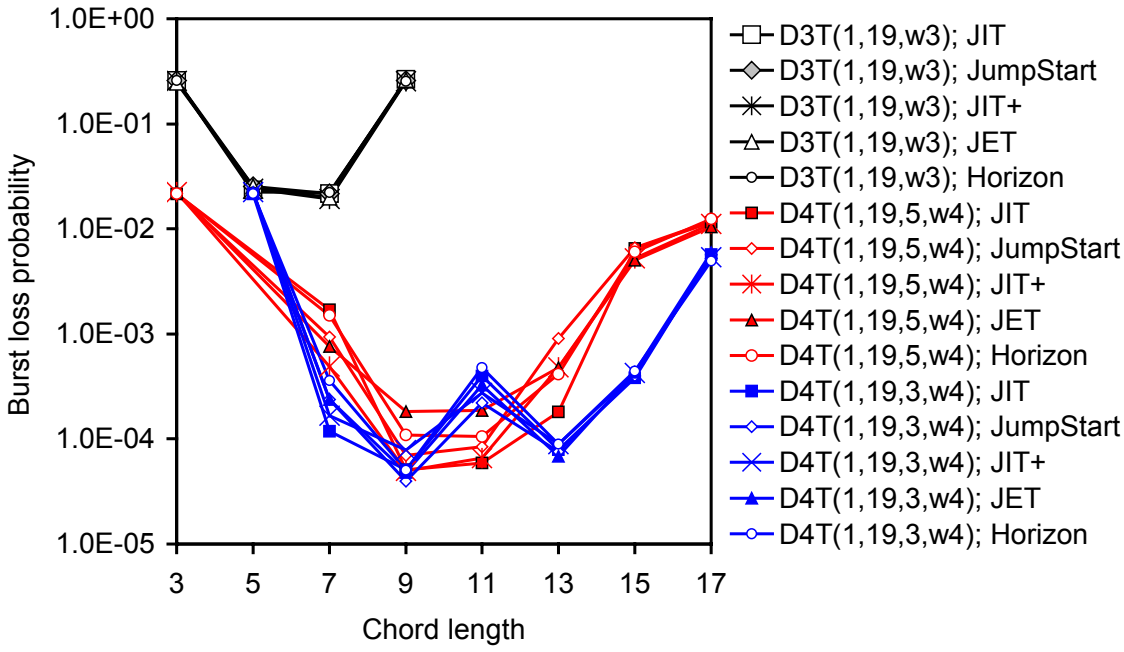


Fig. 4.31. Burst loss probability *versus* chord length for D3T(1,19,w_3), D4T(1,19,3,w_4), and D4T(1,19,5,w_4); N=20; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

Figure 4.32 shows the burst loss probability in the last hop of D2T(1,19), D3T(1,19,7), and D4T(1,19,3,9), for JIT, JumpStart, JIT$^+$, JET, and Horizon resource reservation protocols. As may be seen in this figure, chordal rings clearly have better performance than rings, and, in terms of chordal rings, degree-four topologies have better performance than degree-free. For each D4T(1,19,3,9) and D4T(1,19,5,9) topologies, the burst loss probabilities obtained for JIT, JumpStart, JIT$^+$, JET and Horizon are very similar, as may be seen in Figure 4.33.

Figure 4.34 shows the burst loss probability, for N=20 nodes, for D2T(1,19), D3T(1,19,7), and D4T(1,19,3,9) as a function of the number of hops, for JIT, JumpStart, JIT$^+$, JET and Horizon, and Figure 4.35 shows the burst loss probability for D4T(1,19,3,9) and D4T(1,19,5,9). These figures clearly confirm the results of Figures 4.32 and 4.33, i.e. degree-three chordal rings clearly have better performance than rings, and degree-four chordal rings clearly perform better than degree-three chordal ring networks.

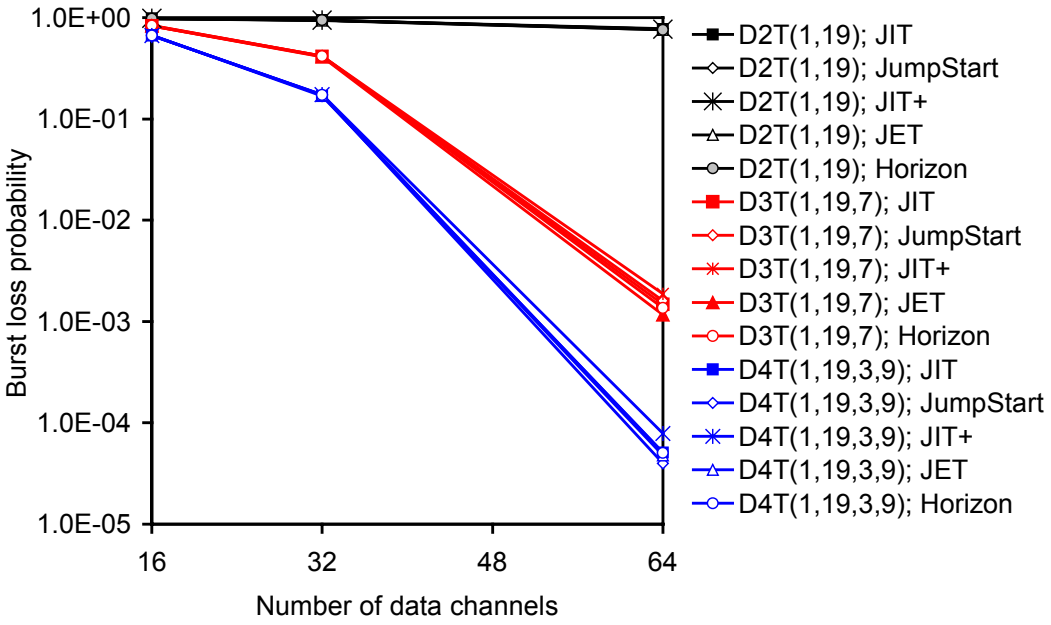


Fig. 4.32. Burst loss probability in the last hop of each topology *versus* number of data channels for D2T(1,19), D3T(1,19,7), and D4T(1,19,3,9); N=20; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

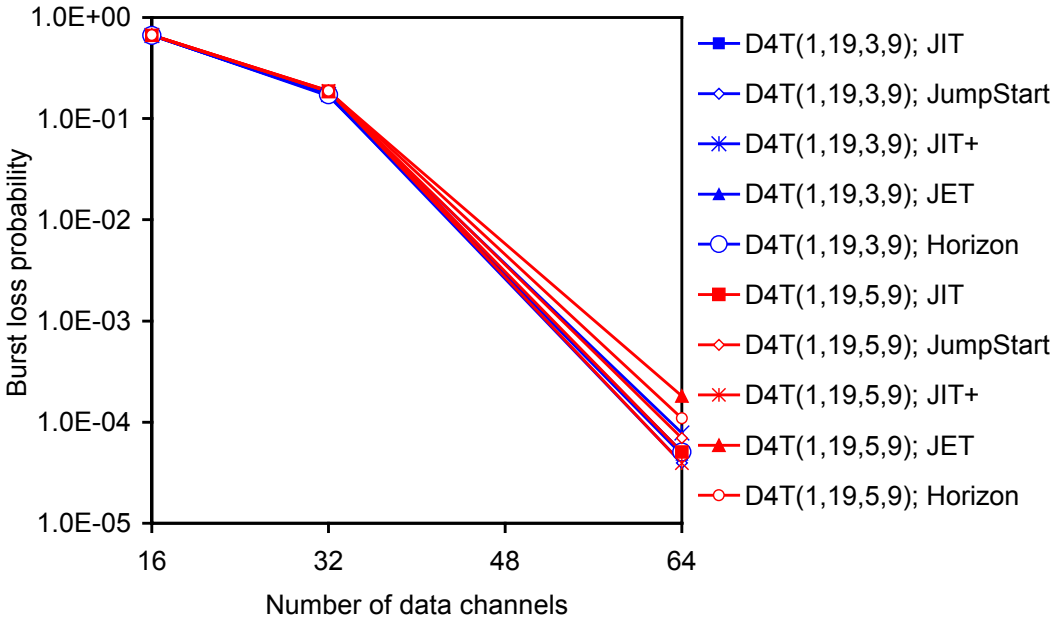


Fig. 4.33. Burst loss probability in the last hop of each topology *versus* number of data channels for D4T(1,19,3,9) and D4T(1,19,5,9); N=20; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

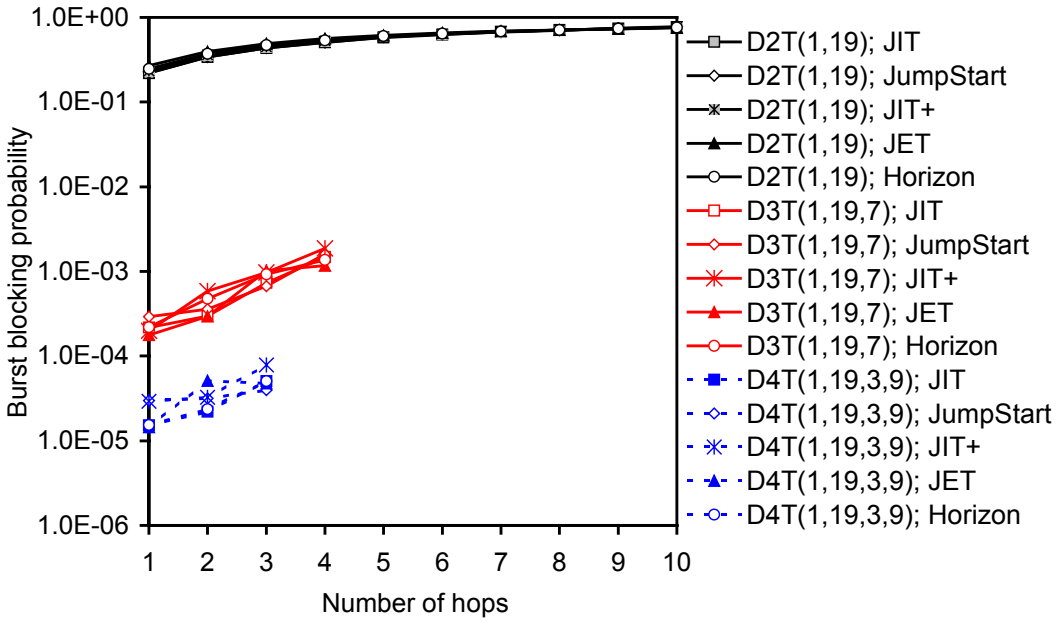


Fig. 4.34. Burst loss probability *versus* number of hops for D2T(1,19), D3T(1,19,7), and D4T(1,19,3,9); N=20; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

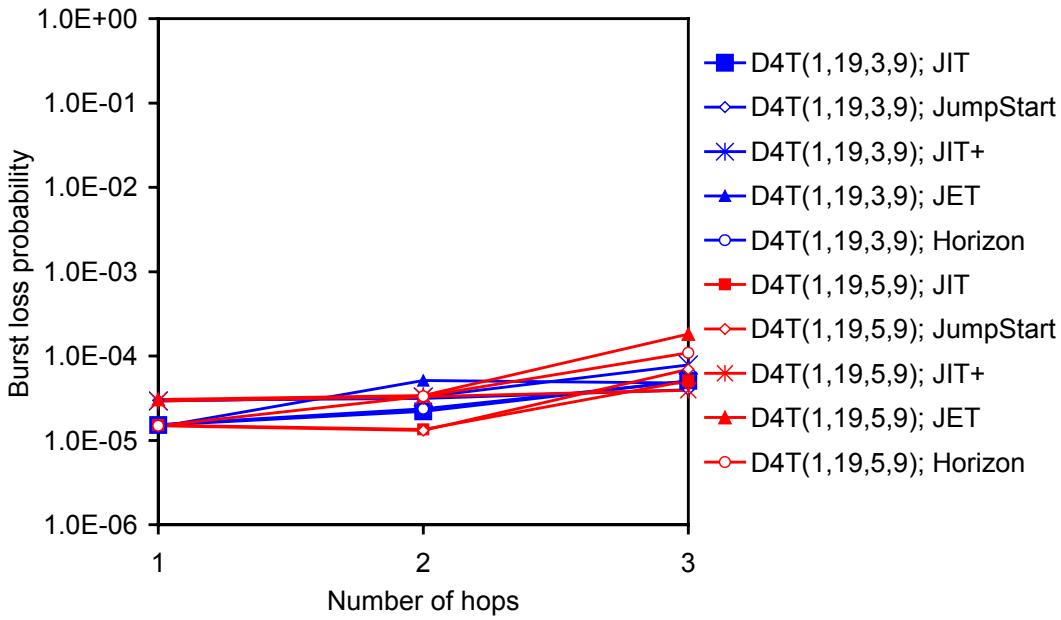


Fig. 4.35. Burst loss probability *versus* number of hops for D4T(1,19,3,9) and D4T(1,19,5,9); N=20; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

4.4.2 Study of Nodal Degree Gain

This sub-section focus the behavior of nodal degree gain obtained in chordal ring network topologies regarding the correspondent ring with the same number of nodes. The nodal degree gain was previously defined in Section 4.2.

Figure 4.36 shows the nodal degree gain, $G_{n,k}(i,j)$ in the last hop of each topology, due to the increase of nodal degree from 2 (D2T(1,19)) to 3 (D3T(1,19,7)), and from 2 (D2T(1,19)) to 4 (D4T(1,19,3,9)) for JIT, JumpStart, JIT$^+$, JET, and Horizon protocols (F=64). For F=16 and F=32 the nodal degree gain is very small due to the high burst loss probability. However, when the number of data channels per link increases to 64, a nodal degree gain between one and two orders of magnitude is observed for degree-three chordal rings. Furthermore, the increase of nodal degree from 2 (D2T(1,19)) to 4 (D4T(1,19,3,9)) leads to a performance improvement between 3 and 4 orders of magnitude. Another important observation that can be made from this figure is that the five resource reservation protocols under study lead to very similar nodal degree gains both when the nodal degree increases from 2 to 3 ($G_{2,3}$(10,4)) and when the nodal degree increases from 2 to 4 ($G_{2,4}$(10,3)).

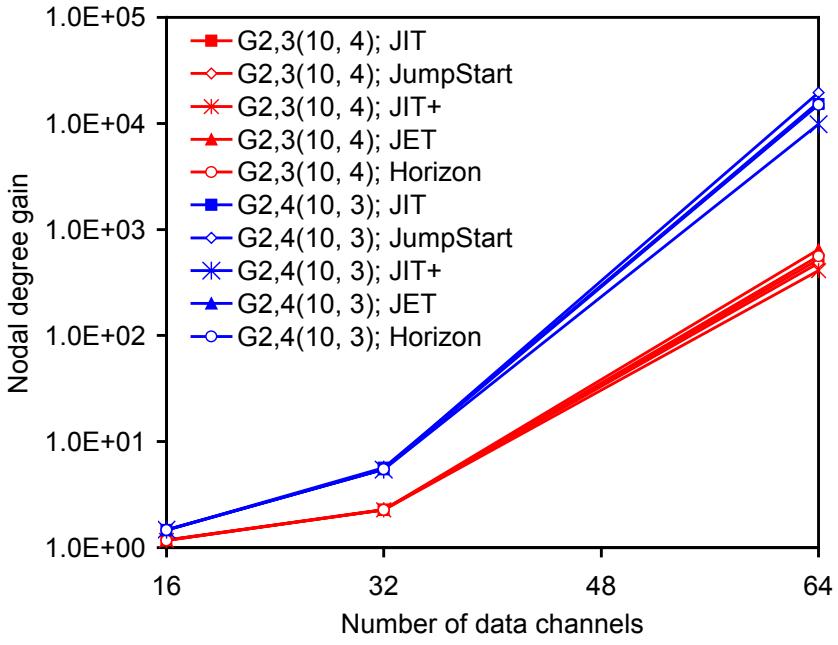


Fig. 4.36. Nodal degree gain, in the last hop of each topology, due to the increase of the nodal degree from 2 (D2T(1,19)) to 3 (D3T(1,19,7)) and to 4 D4T(1,19,3,9); N=20; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

4.4.3 Effects of Setup Message Processing Time and OXC Configuration Time

This sub-section discusses the influence of the setup message processing time and the optical cross-connect (OXC) configuration time on the performance of degree-four chordal ring topologies, for JIT, JumpStart, JIT$^+$, JET, and Horizon protocols. The network with the smallest diameter selected for degree-four network topologies is D4T(1,19,3,9) which presents the best performance. Degree-three chordal ring topologies are also considered for comparison purposes.

Figure 4.37 shows the burst loss probability as a function of OXC configuration time in the last hop of D3T(1,19,7) and D4T(1,19,3,9) for F=64 and λ/μ=44.8. In this figure, T_{Setup} time is defined for JIT, JumpStart, and JIT$^+$ and it is estimated for JET and Horizon taking in account the current available technology (JITPAC controllers [141]). T_{OXC} is assumed to range from the value estimated for a near future scenario (T_{OXC} =20μs) up to ten times the value defined for currently available technology, i.e. T_{OXC} =10*10ms=100ms. A figure with the same network conditions but with λ/μ=32 is not shown because the result of the burst loss probability for D4T(1,19,3,9) when T_{OXC}<10ms tends to zero. As may be seen in Figure 4.37, degree-four chordal ring topology clearly has better performance than degree-three, mainly, for $T_{OXC}\leq$50ms. It may also be observed that for $T_{OXC}\leq$1ms, the performance of the five protocols is more or less constant. In D4T(1,19,3,9), for $T_{OXC}\leq$1ms, the performance of different protocols presents some oscillations. Even so, it is possible to conclude that despite the improvement and development of new technologies, the network does not present a better performance, and T_{OXC} (for values less than 1ms) does not influence the performance of those networks. However, when T_{OXC}>1ms, it is possible to observe that the value of burst loss probability increases with the increasing of the value of T_{OXC} from 1ms to 100ms. This behavior expresses the amount of time that the resources of OXC are reserved for a burst. The results expressed in the Figure 4.37 are confirmed by Figure 4.38, which plots the burst loss probability as a function of OXC configuration time in the last hop of D3T(1,19,7) and D4T(1,19,3,9) for the five protocols under study, with F=64, λ/μ=44.8. T_{Setup} is assumed to change with T_{OXC}, according to (4.3), (4.4), and (4.5). Again, values of T_{OXC} smaller than 1ms do not have a significant impact.

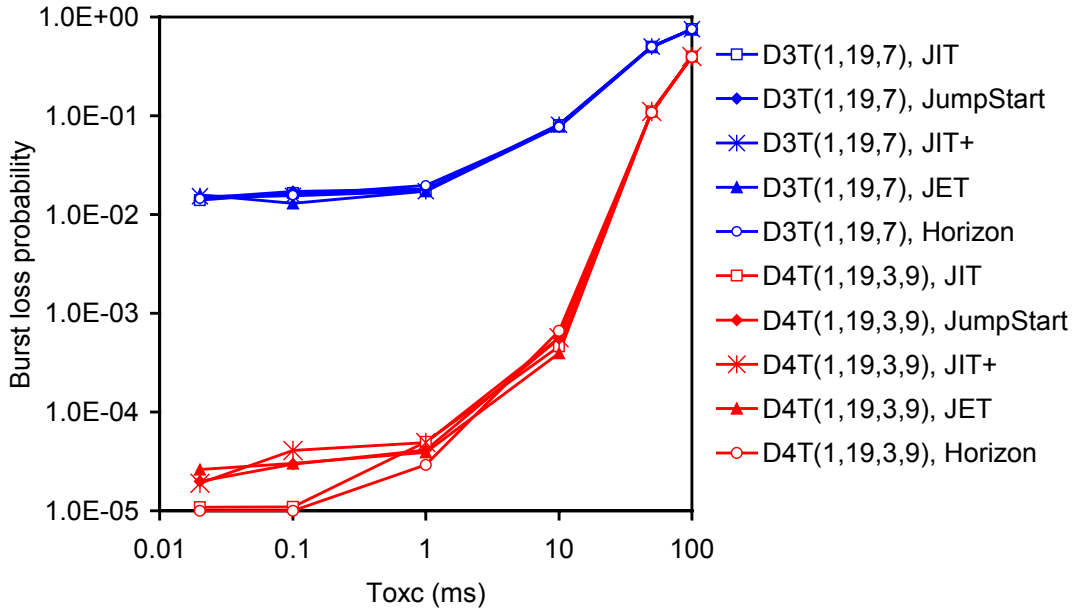


Fig. 4.37. Burst loss probability as a function of OXC configuration time in the last hop of D3T(1,19,7) and D4T(1,19,3,9) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; λ/μ=44.8; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

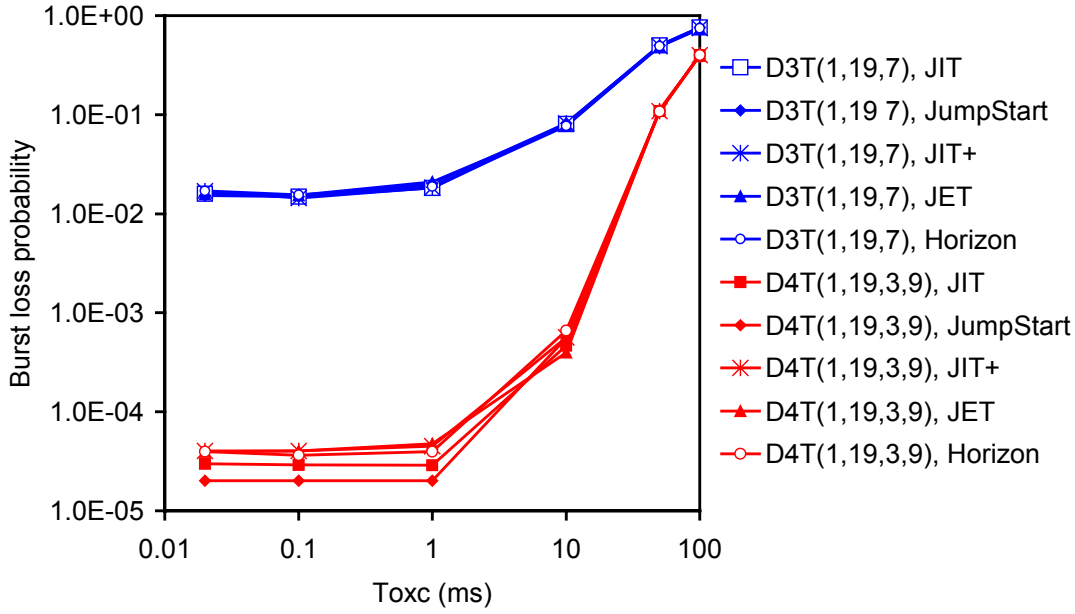


Fig. 4.38. Burst loss probability as a function of OXC configuration time in the last hop of D3T(1,19,7) and D4T(1,19,3,9) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; λ/μ=44.8; with changing T_{Setup} according to (4.3), (4.4), and (4.5) for each resource reservation protocol.

Figure 4.39 illustrates the burst loss probability as a function of setup message processing time (T_{Setup}) in the last hop of D3T(1,19,7) and D4T(1,19,3,9) for JIT, JumpStart, JIT$^+$, JET, and Horizon, with F=64 and λ/μ=44.8. Two scenarios are considered regarding T_{OXC}: it assumes the value for the currently available technology (T_{OXC}=10ms) or an estimated value for a near future scenario (T_{OXC}=20μs). For each curve of Figure 4.39, T_{OXC} is assumed to have a fix value while T_{Setup} ranges between the values considered for the current available technology and the estimated values for the near future technology. Thus, T_{Setup} ranges between 12.5μs and 1μs, for JIT, JumpStart, and JIT$^+$, it ranges between 25μs and 2μs for JET, and between 50μs and 4μs for Horizon. A figure with the same network conditions but with λ/μ=32 is not shown because the result of the burst loss probability for D4T(1,19,3,9) tends to zero for values of T_{OXC} smaller than 10ms. As may be seen, the performance of degree-four chordal rings is clearly better than the degree-three and the behavior of the five protocols is very close. This figure shows that the reduction of T_{Setup} performs slightly better for degree-four topology, mainly, for T_{OXC}=20μs. On the other hand, the reduction of OXC configuration time performs better. Furthermore, the influence of the nodal degree in the performance of a given topology is larger than the influence of the T_{OXC}, as may be seen between degree three and degree four network topologies. It may also be observed that for degree-four chordal rings a reduction of the T_{OXC} from 10 ms down to 20 μs leads to a performance improvement of about three orders of magnitude. For degree-three chordal rings, the reduction of T_{OXC} leads to a performance improvement of about two orders of magnitude.

Figure 4.40 plots the burst loss probability as a function of Setup message processing time in the last hop of D3T(1,19,7) and D4T(1,19,3,9) for JIT, JumpStart, JIT$^+$, JET, and Horizon, being T_{OXC} computed according to (4.6), (4.7), and (4.8) for each protocol. As may be seen, when the value of T_{Setup} increases, the correspondent burst loss probability also increases, being around one order of magnitude for degree-four chordal ring. This figure confirms previous results in terms of the best performance of degree-four chordal rings in comparison with degree-three, being between two and three orders of magnitude, and it also proves the influence of T_{OXC} and T_{Setup} in the performance of OBS networks.

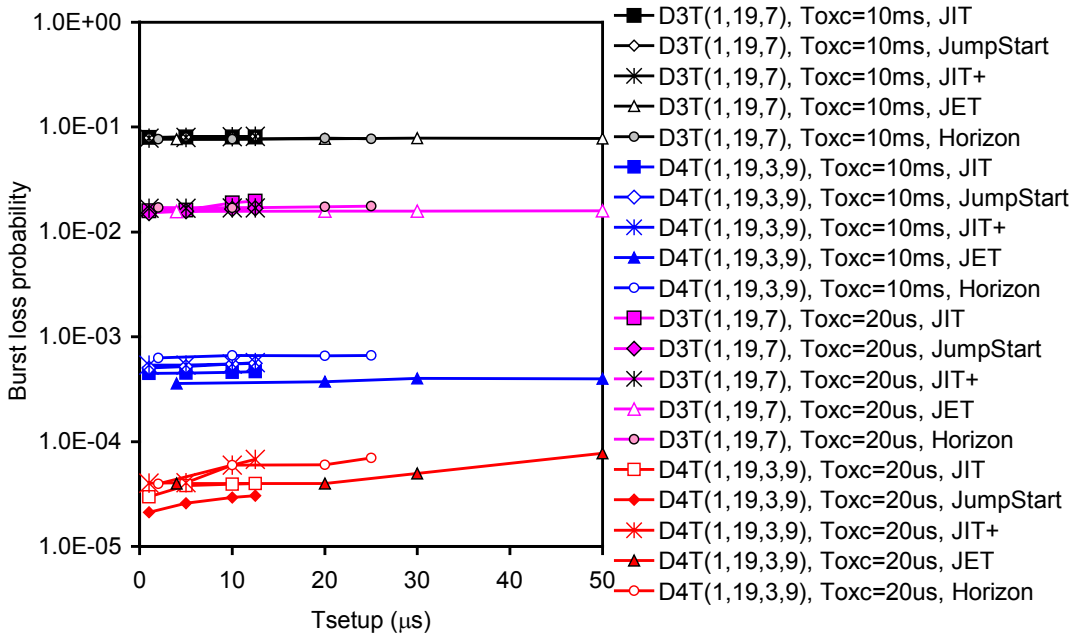


Fig. 4.39. Burst loss probability as a function of setup message processing time in the last hop of D3T(1,19,7) and D4T(1,19,3,9) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; N=20; λ/μ=44.8; T_{OXC}=10ms; T_{OXC}=20μs.

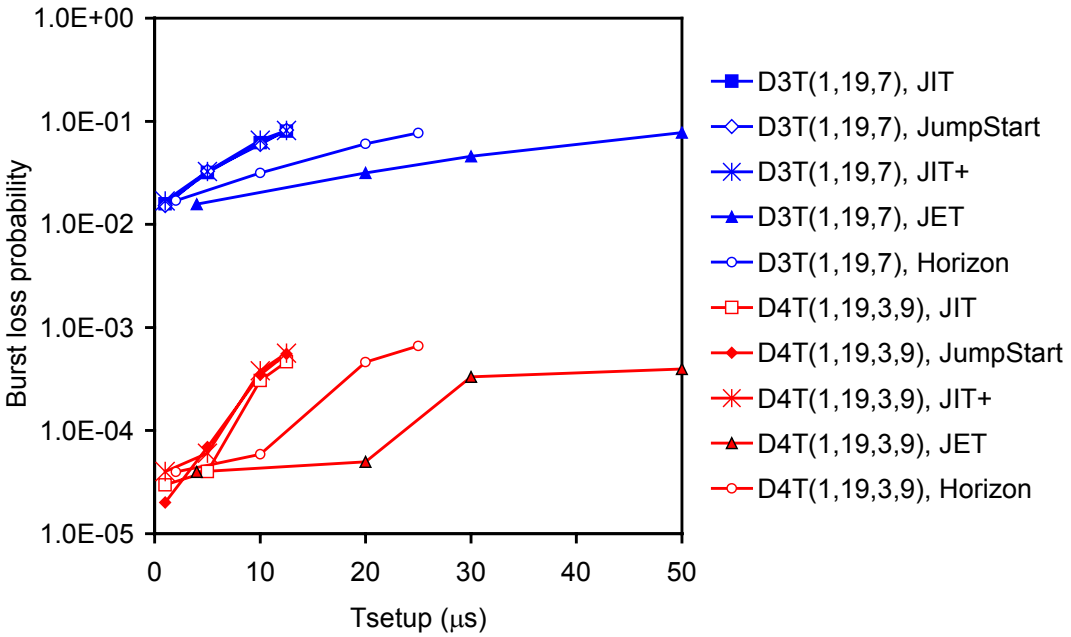


Fig. 4.40. Burst loss probability *versus* T_{Setup} in the last hop of D3T(1,19,7) and D4T(1,19,3,9) for JIT, JumpStart, JIT+, JET, and Horizon; F=64; N=20; λ/μ=44.8; with changing T_{OXC} according to (4.6), (4.7), and (4.8) for each resource reservation protocol.

4.5 Conclusions

A performance analysis of optical burst switched degree-three and degree-four chordal ring networks was presented for JIT, JumpStart, JIT$^+$, JET and Horizon resource reservation protocols. For comparison purposes, ring topologies were also considered. The burst loss probability was analyzed for OBS networks with D2T(1,19), D3T(1,19,3), D3T(1,19,5), D3T(1,19,7), D3T(1,19,9), D4T(1,19,5,9), and D4T(1,19,3,9).

First, the burst loss probability, the nodal degree gain, and the chord length gain were defined. Burst loss probability is a very important performance metric to evaluate the performance of OBS networks and it was defined as the probability of a burst transmission does not arrive at its destination. Nodal degree gain was proposed in order to quantify the benefits due to the increase of nodal degree while the chord length gain allows to quantify the benefits due to a better choice of the chord length.

After, the study was focused on the performance of degree-three chordal ring networks, and it was shown that for a network with 20 nodes, the best performance was obtained for D3T(1,19,7). It was revealed that the nodal degree gain due to the increase of nodal degree from two (D2T(1,19)) to three (D3T(1,19,7)) is about three orders of magnitude in the first hop of both topologies, and is between two and three orders of magnitude in the last hop of each topology. It was also shown that the largest chord length gain is slightly less than two orders of magnitude, due to the choice of the chord length of w_3=7 instead of a chord length of w_3=3. Concerning the study of the effect of the setup message processing time (T_{Setup}) and the OXC configuration time (T_{OXC}), it was observed that for $T_{OXC}\leq$1ms, the performance of the chordal ring is independent of the change of the T_{OXC}, which means that a reduction of the values of T_{OXC} to ones smaller than 1ms does not improve the network performance. It was also observed that for chordal rings a reduction of the T_{OXC} from 10ms down to 20μs leads to a performance improvement of about two orders of magnitude. For rings, the burst loss is so high that the reduction of T_{OXC} does not have impact on the network performance. Another important conclusion is that the network performance for JIT, JumpStart, JIT$^+$, JET and Horizon resource reservation protocols is very close.

Another issue analyzed in this chapter was the performance assessment of OBS degree-three and degree-four chordal ring networks for the resource reservation protocols under study. For comparison purposes, ring topologies are also considered. It was shown that, for a network with 20 nodes, degree-four chordal ring topologies with smallest diameter have better performance regarding other topologies with the same nodal degree and different chord length (w_4). The nodal degree gain due to the increase of nodal degree from two (ring) to three (degree-three chordal ring) is between two and three orders of magnitude in the last hop of each topology. It was also shown that the nodal degree gain due to the increase of nodal degree from two to four (degree-four chordal ring) is around four orders of magnitude. The effect of setup message processing time (T_{Setup}) and OXC configuration time (T_{OXC}) was analyzed. The reduction of T_{Setup} performs slightly better for degree-four topology, mainly, for T_{OXC}=20μs, and with the reduction of T_{OXC} the network performs better. It was observed that the influence of the nodal degree in the performance of a given chordal ring topology is larger than the influence of T_{Setup} and T_{OXC}. This study confirms previous observations concluding that, for 20 nodes, degree-four chordal ring performs better than degree-three and the worst performance is presented by ring. It was observed in all cases studied that the performance of the five resource reservation protocols under study is very similar.

Chapter 5

Performance Assessment of OBS Mesh Networks for One-way Resource Reservation Protocols

5.1 Introduction

This chapter focuses on analysis of OBS networks with the following mesh topologies reported in Chapter 3: chordal rings with number of nodes ranging from 10 up to 30, mesh-torus with 16 and 25 nodes, the NSFNET with 14-nodes and 21 links, the NSFNET with 16 nodes and 25 links, the ARPANET with 20 nodes and 32 links, the European Optical Network (EON) with 19 nodes and 37 links, and the Portuguese *Fundação para a Computação Científica Nacional* network (FCCN-NET) with 14 nodes and 14 links. For comparison purposes bi-directional ring topologies are also considered. These topologies have the following nodal degree: ring: 2.0; chordal ring: 3.0; mesh-torus: 4.0; NSFNET with 14-node and 21 links: 3.0; the NSFNET with 16 nodes and 25 links: 3.125; the ARPANET with 20 nodes and 32 links: 3.2; the EON: 3.895, and the FCCN-NET: 2. Nodal degree is the average number of links connected to each node of a given topology and it is calculated as a function of number of nodes and number of links. For example, one considers a network topology with *N* nodes and *L* links. As these links are bi-directional, the total number of unidirectional

links is 2*L*. Then, dividing the number of unidirectional links – 2*L* - by the number of nodes - *N* -, the nodal degree is equal to 2*L*/*N*.

The first sections of this chapter show a performance assessment of JIT, JumpStart, JIT$^+$, JET, and Horizon resource reservation protocols in OBS networks with mesh topologies, and later sections present a performance evaluation of E-JIT protocol. This performance study is focused on the burst loss probability obtained through the simulator described in Chapter 3. Simulation parameters are the same ones described in Chapter 4 (presented in Section 4.1).

The outline of the chapter is as follows. Section 5.2 studies the impact of network size on the network performance considering mesh topologies with 14 and 16 nodes, and every topology considered in this study with number of nodes between 10 and 30. Section 5.3 discusses the role of nodal degree in OBS mesh networks, considering mesh topologies with 16 and 20 nodes, and the influence of nodal degree on the nodal degree gain considering mesh topologies with nodal degree between three and six. Section 5.4 discusses the impact of setup message processing time and optical cross-connect configuration time in OBS mesh networks. Section 5.5 analyses the performance assessment of E-JIT, in comparison with JIT, taking into account the mesh topologies with 16 and 20 nodes, the impact of number of nodes, the influence of nodal degree on nodal degree gain, and the effect of setup message processing time and OXC configuration time. Section 5.6 concludes the chapter.

This chapter is partially based on papers [23-30, 33, 166].

5.2 Impact of Network Size on the Network Performance

This section discusses the impact of the network size on the network performance using OBS switching paradigm. First, the study is focused on mesh network topologies with 14 and 16 nodes and, after, it makes a performance comparison considering mesh networks with number of nodes between 10 and 30.

5.2.1 Performance Assessment of Networks with 14 Nodes

This sub-section is focused on the study of OBS networks considering mesh topologies with 14 nodes to compare them with the network topology proposed for

the Portuguese *Fundação para a Computação Científica Nacional* (FCCN) [156], presented in 3.6.6. The proposed FCCN network (FCCN-NET) has 14 nodes and 14 links (Figure 3.12).

Figure 5.1 shows the burst loss probability as a function of number of data channels per link (F), in the last hop of ring, degree-three (D3T(w_1, w_2, w_3)), NSFNET, and FCCN-NET networks. As may be seen, when enough network resources are available (F=64), the degree-three chordal ring networks with chord length of w_3=5 clearly have better performance. The FCCN-NET and ring topologies (with same number of nodes and links) present a similar behavior in terms of burst loss probability. Another outcome that was found was the close performance of the different resource reservation protocols, independently of the considered topology. Figure 5.2 confirms this result for a different number of hops. Since the burst loss probability is a major issue in OBS networks, clearly ring and FCCN-NET topologies are the worst choice for this kind of networks due to very high loss probabilities and, surprisingly, degree-three chordal rings with smallest diameter have a very good performance with burst loss probabilities ranging from 10^{-2}-10^{-4}, depending on the number of hops.

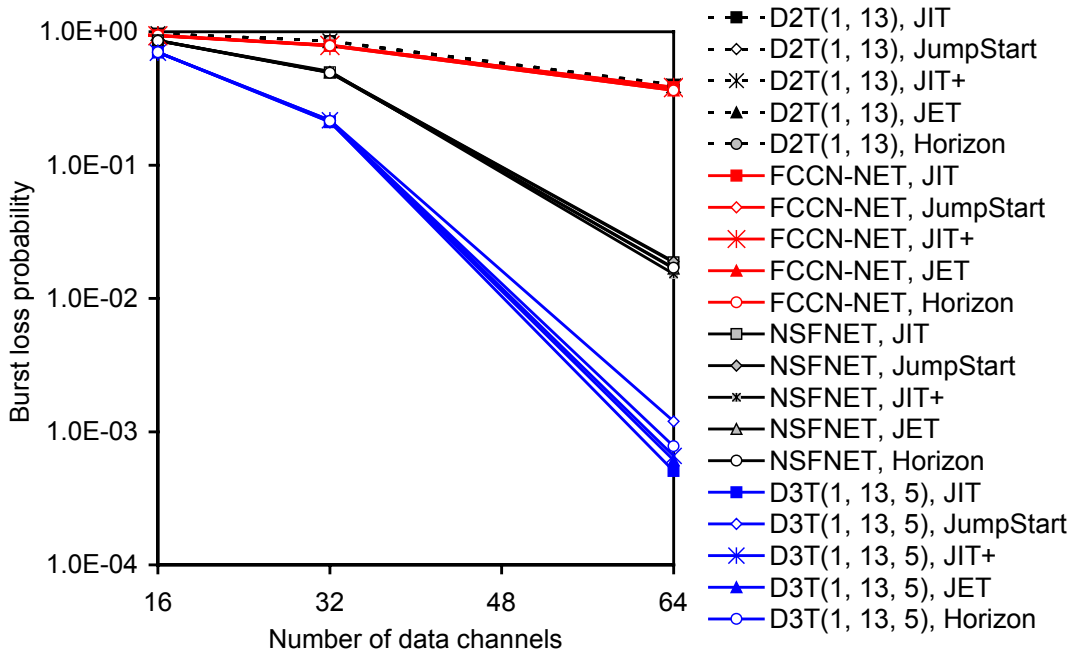


Fig. 5.1. Burst loss probability, as a function of the number of data channels per link (F), in the last hop of D2T(1,13), FCCN-NET, NSFNET, and D3T(1,13,5) for JIT, JumpStart, JIT$^+$, JET, and Horizon; N=14; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

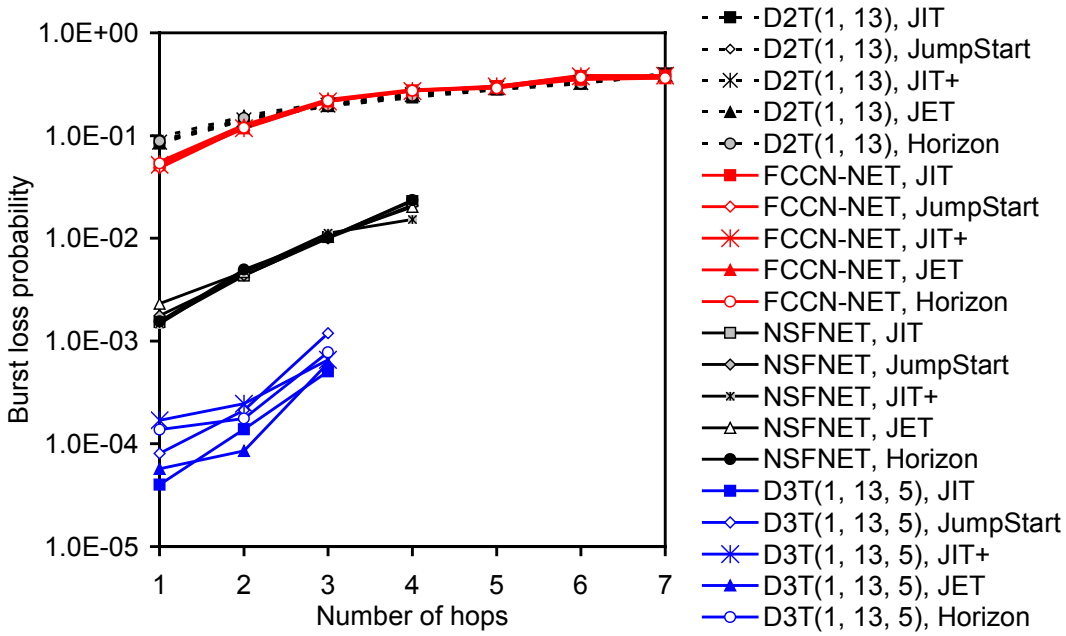


Fig. 5.2. Burst loss probability, as a function of the number of hopes, for D2T(1,13), FCCN-NET, NSFNET, and D3T(1,13,5), for JIT, JumpStart, JIT$^+$, JET, and Horizon; N=14; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

5.2.2 Performance Assessment of Networks with 16 Nodes

This sub-section is devoted to study a performance assessment of OBS networks for several regular and irregular mesh topologies with N=16 nodes. Rings, chordal rings and mesh-torus topologies are examples of regular topologies while NSFNET is an example of an irregular topology.

Figure 5.3 shows the burst loss probability in the last hop of ring, chordal rings, mesh-torus and NSFNET networks, all with 16 nodes. As may be seen in Figure 5.3, when enough network resources are available (F=64), the chordal ring networks with chord length of w_3=5 clearly have better performance. This figure also shows that the performance of the NSFNET is very close to the performance of chordal rings with chord length of w_3=3 or w_3=7. These results reveal the importance of the way links

are connected in the network, since chordal rings and NSFNET have similar nodal degrees and therefore a similar number of network links. Also interesting is the fact that chordal rings with w_3=5 (D3T(1, 15, 5)) have better performance than mesh-torus networks, which have a nodal degree of 4, i.e., with 25% more of network links. It was also possible to observe that the best performance of chordal ring network is obtained for the smallest network diameter. Results presented in Figure 5.3 were obtained for the JIT resource reservation protocol. Similar results have been obtained for JumpStart, JIT$^+$, JET, and Horizon, showed in Figures 5.4, 5.5, 5.6, and 5.7, respectively.

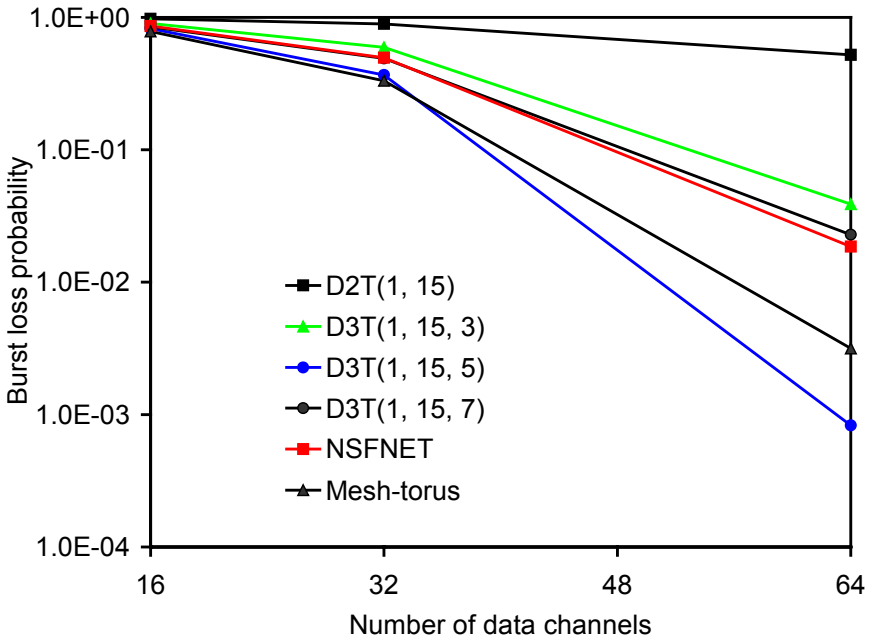


Fig. 5.3. Burst loss probability, as a function of the number of data channels per link (F), in last hop of ring, chordal rings, NSFNET and mesh-torus networks for JIT; N=16; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

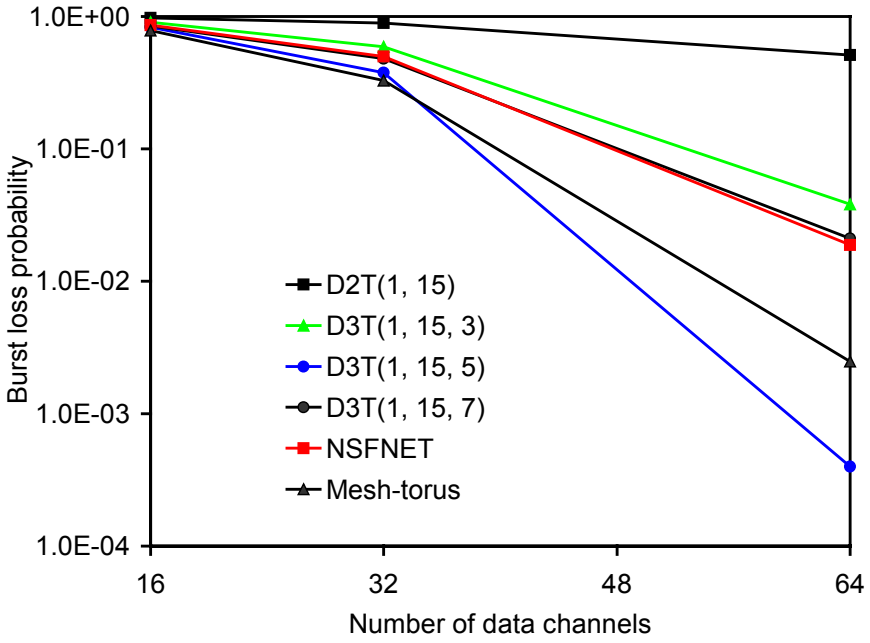


Fig. 5.4. Burst loss probability, as a function of the number of data channels per link (F), in last hop of ring, chordal rings, NSFNET and mesh-torus networks for JumpStart; N=16; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

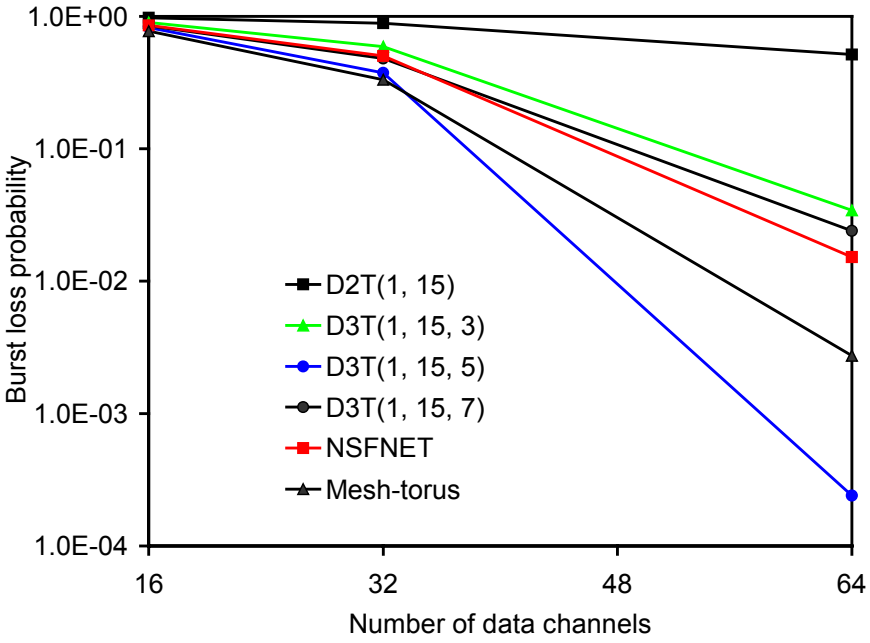


Fig. 5.5. Burst loss probability, as a function of the number of data channels per link (F), in last hop of ring, chordal rings, NSFNET and mesh-torus networks for JIT$^+$; N=16; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

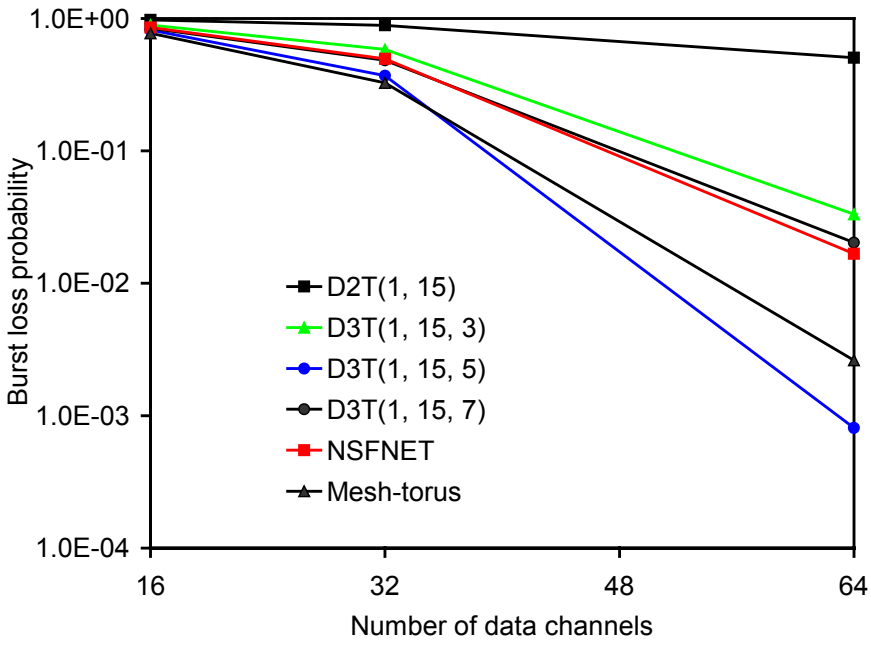


Fig. 5.6. Burst loss probability, as a function of the number of data channels per link (F), in last hop of ring, chordal rings, NSFNET and mesh-torus networks for JET; N=16; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

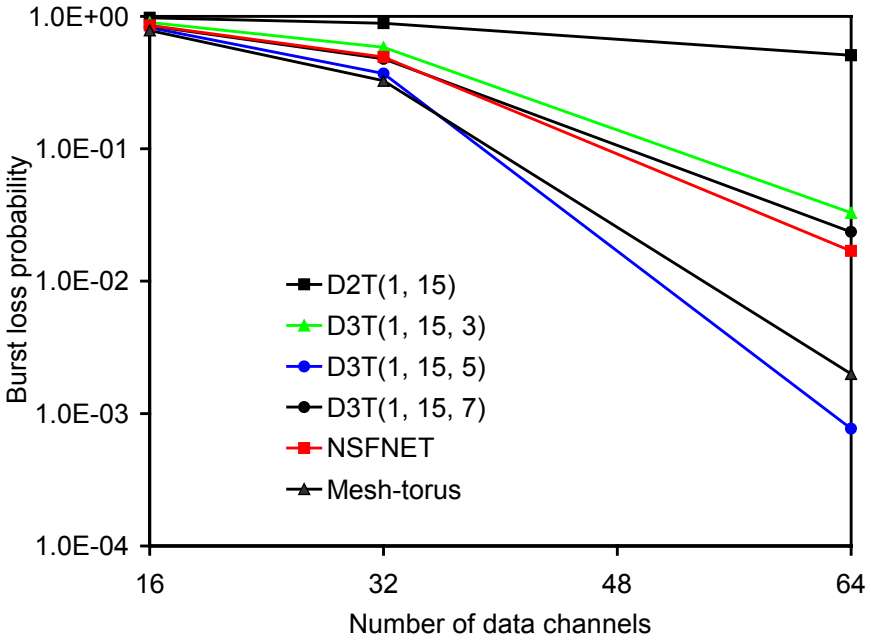


Fig. 5.7. Burst loss probability, as a function of the number of data channels per link (F), in last hop of ring, chordal rings, NSFNET and mesh-torus networks for Horizon; N=16; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

Figures 5.8 and 5.9 show the burst loss probability in the last hop of ring, degree-three (D3T(w_1, w_2, w_3)) and degree-four (D4T(w_1, w_2, w_3, w_4)) chordal rings, NSFNET, and mesh-torus networks, all with 16 nodes. As may be seen in Figure 5.8, when enough network resources are available (F=64), the degree-three chordal ring network with chord length of w_3=5 clearly have better performance. Among degree-three chordal rings, it is possible observe that the chord lengths that led to smallest network diameters also led to the best network performance. In Figures 5.8 and 5.9, only chordal rings topologies with smallest diameters are considered, i.e. D3T(1, 15, 5) and D4T(1, 15, 5, 9). One observed that the performance of the NSFNET is very close to the performance of degree-three chordal rings with chord lengths of w_3=3 or w_3=7, which are not chordal rings with smallest diameter, and therefore they are not shown in Figures 5.8 and 5.9. These results reveal the importance of the way links are connected in the network, since chordal rings and NSFNET have similar nodal degrees and therefore a similar number of network links. Also interesting is the fact that degree-three chordal rings with w_3=5 have better performance than mesh-torus networks, which have a nodal degree of 4, i.e., mesh-torus networks have 25% more of links than degree-three chordal rings. Another interesting result, for 16 nodes, is that the addition of a new chord to the degree-three chordal ring led to a small improvement in the network performance. Results presented in Figures 5.8 and 5.9 were obtained for the JIT, JumpStart, JIT$^+$, JET, and Horizon protocols, and, as may be seen, their performance is very close, independently of the considered topology. Figures 5.10 and 5.11 confirm these results for a different number of hops. Since the burst loss probability is a major issue in OBS networks, clearly ring topologies are the worst choice for this kind of networks due to very high burst loss probabilities and, surprisingly, degree-three chordal rings with smallest diameter have a very good performance with burst loss probabilities ranging from 10^{-3}-10^{-5}, depending on the number of hops.

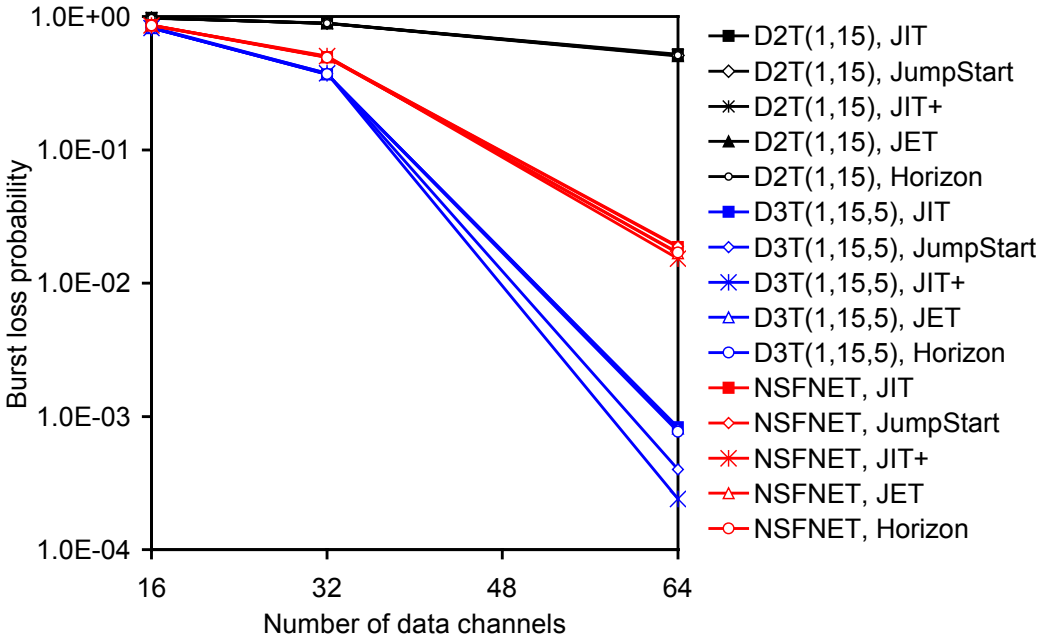


Fig. 5.8. Burst loss probability, as a function of the number of data channels per link (F), in the last hop of D2T(1,15), D3T(1,15,5), and NSFNET for JIT, JumpStart, JIT$^+$, JET, and Horizon; N=16; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs.

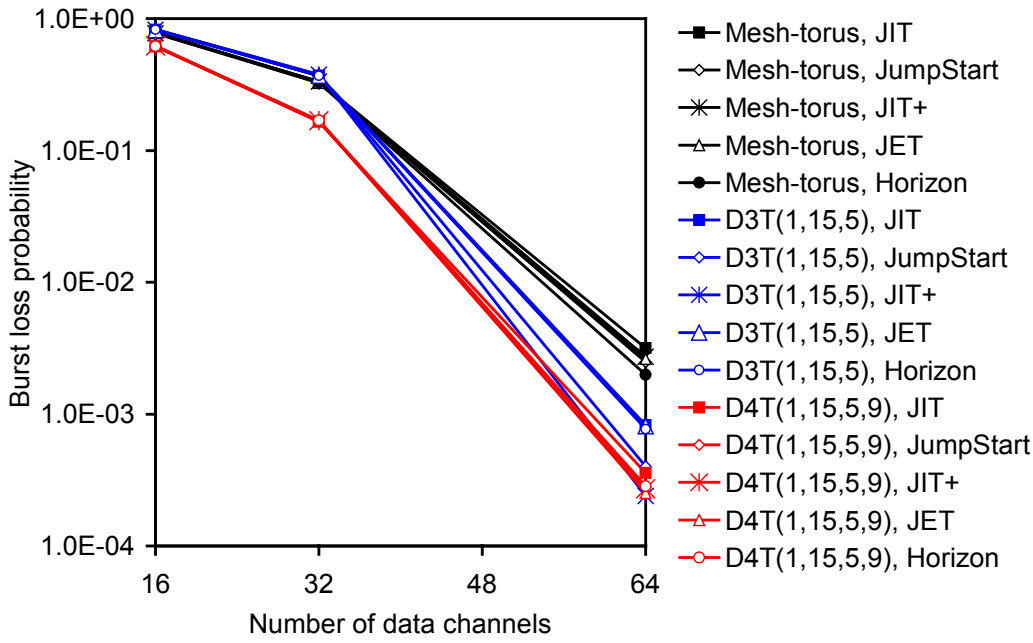


Fig. 5.9. Burst loss probability, as a function of the number of data channels per link (F), in the last hop of D3T(1,15,5), Mesh-torus, and D4T(1,15,5,9) for JIT, JumpStart, JIT$^+$, JET, and Horizon; N=16; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs.

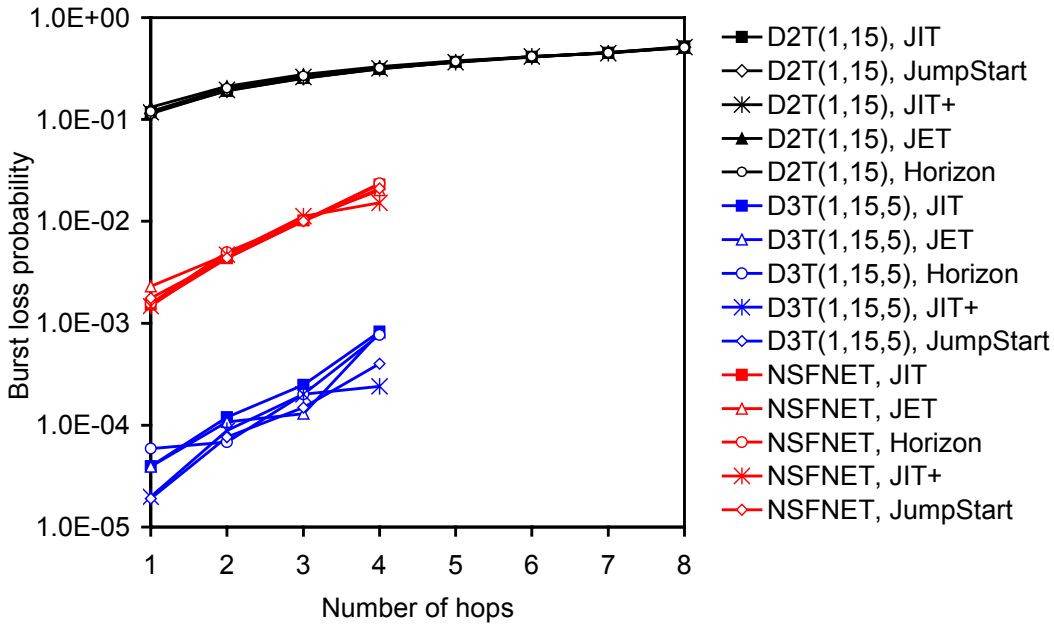


Fig. 5.10. Burst loss probability, as a function of the number of data channels per link (F), in the last hop of D3T(1,15,5), Mesh-torus, and D4T(1,15,5,9) for JIT, JumpStart, JIT$^+$, JET, and Horizon; N=16; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs.

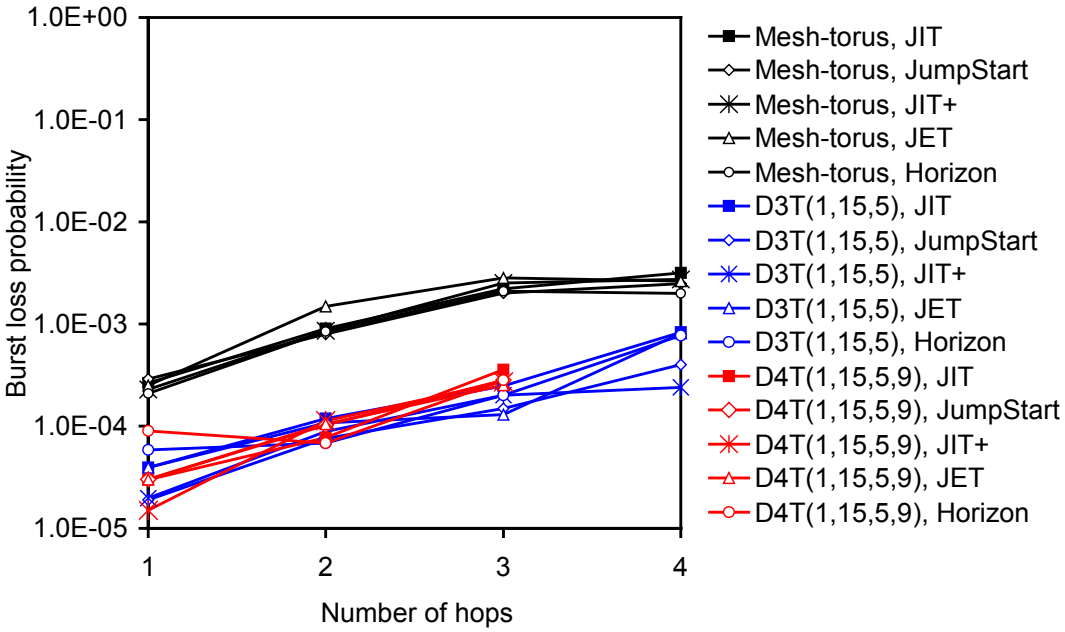


Fig. 5.11. Burst loss probability, as a function of the number of hops, for degree-three and degree-four chordal rings (D3T(1,15,5) and D4T(1,15,5,9)), and Mesh-torus networks using JIT, JumpStart, JIT$^+$, JET, and Horizon; N=16; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs.

5.2.3 Performance Assessment for Number of Nodes Ranging from 10 up to 30

Figure 5.12 shows the burst loss probability, as a function of the number of nodes (*N*), between *N*=10 and *N*=30, in the last hop of rings, degree-three and degree-four chordal rings, FCCN-NET, NSFNET, ARPANET, EON, and mesh-torus networks, for JIT protocol. Similar results have been obtained for Jumpstart, JIT$^+$, JET, and Horizon, showed in Figures 5.13, 5.14, 5.15, and 5.16, respectively. However, all those results confirm the similar performance of one-way resource reservation protocols. Since the performance of these protocols is similar, protocols with immediate reservation (JIT, JIT$^+$, and JumpStart) are more suitable for OBS mesh networks, because their implementation is simpler than the implementation of protocols with delayed reservation. Figure 5.17 shows the corresponding network diameters and Figure 5.18 presents the network diameters as a function of chord length. As may be seen, when the number of nodes is smaller, the D3T(1,*N*-1,5) has the best performance. When the number of nodes is larger, i.e., more than 24 nodes, D3T(1,*N*-1,7) has better performance than D3T(1,*N*-1,5). However, if the number of nodes is larger than 16, D4T(1,*N*-1,5,9) has the best performance because this topology leads to a smallest diameter in the whole range from 14 to 30 nodes, as Figure 5.17 and Figure 5.18 show. As may be seen, D4T(1,*N*-1,5,9) has better performance for *N*=22 nodes than for other number of nodes. This observation may be explained by the smallest network diameter presented for this topology with *N*=22 nodes in Figure 5.17. The worst performance is verified for ring topologies (D2T(1,*N*-1)) and FCCN-NET. The performance of both mesh-torus (with *N*=16 and *N*=25) is very similar. This behavior may be justified by the same nodal degree of these topologies. NSFNET with *N*=16 nodes has better performance than NSFNET with *N*=14 nodes, although network diameter of NSFNET with *N*=14 nodes (equal to 3) is less than the network diameter of NSFNET with *N*=16 nodes (equal to 4). As may be observed, the performance of ARPANET and EON is lower and very similar.

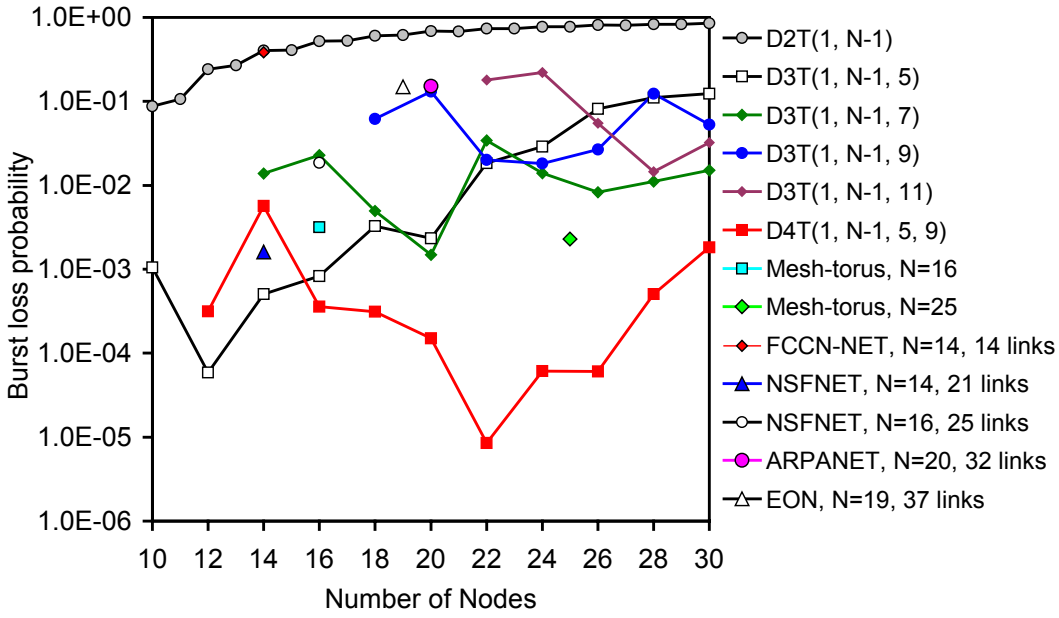


Fig. 5.12. Burst loss probability, as a function of the number of nodes (N), in the last hop of rings, degree-three, and degree-four chordal rings, FCCN-NET, NSFNET, ARPANET, European Optical Network (EON), and mesh-torus, for JIT; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

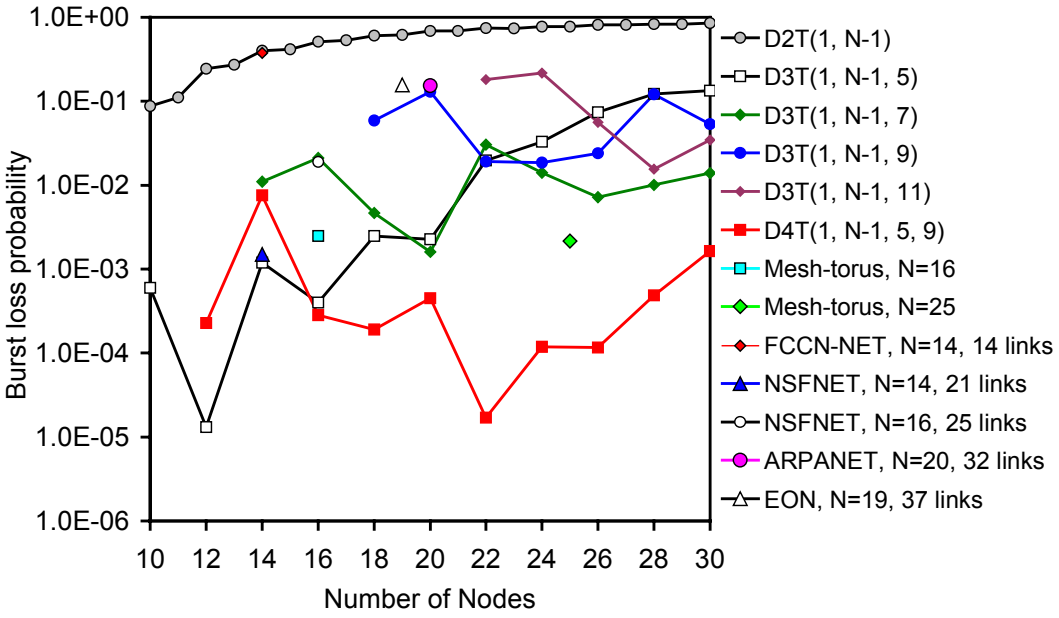


Fig. 5.13. Burst loss probability, as a function of the number of nodes (N), in the last hop of rings, degree-three, and degree-four chordal rings, FCCN-NET, NSFNET, ARPANET, European Optical Network (EON), and mesh-torus, for JumpStart; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

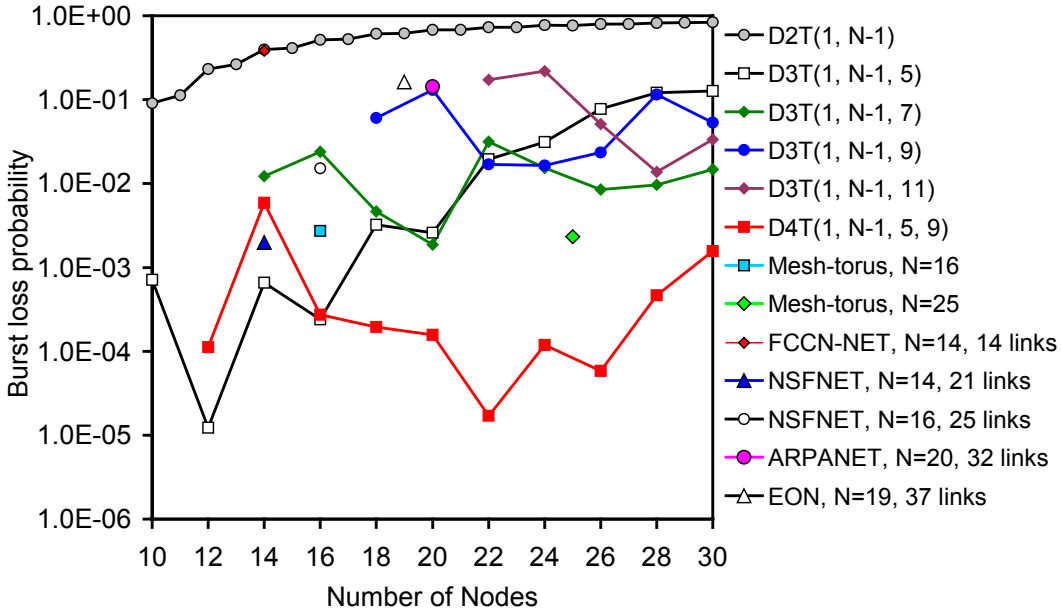


Fig. 5.14. Burst loss probability, as a function of the number of nodes (N), in the last hop of rings, degree-three, and degree-four chordal rings, FCCN-NET, NSFNET, ARPANET, European Optical Network (EON), and mesh-torus, for JIT$^+$; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

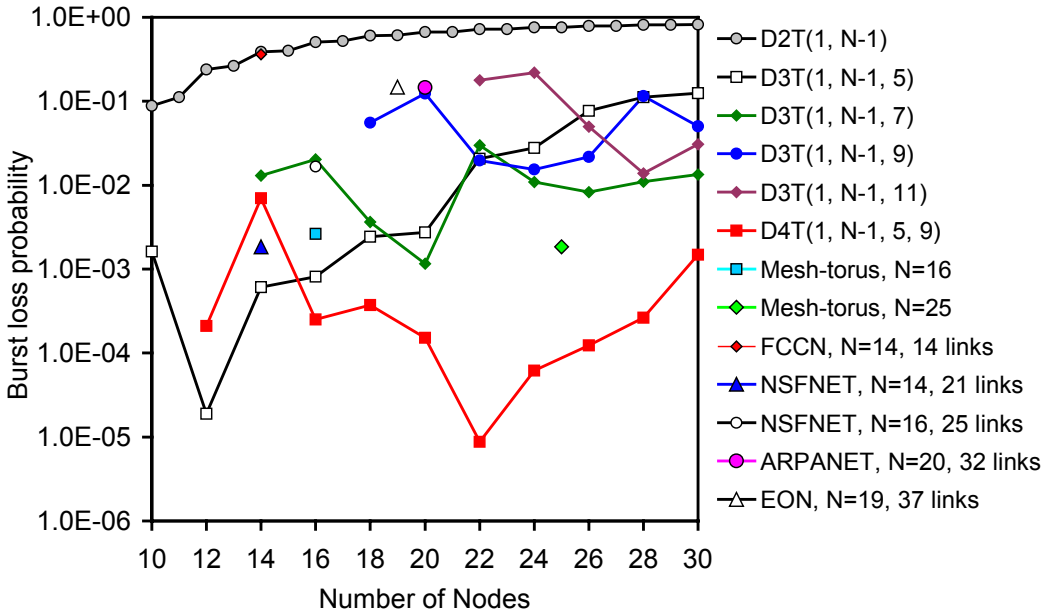


Fig. 5.15. Burst loss probability, as a function of the number of nodes (N), in the last hop of rings, degree-three, and degree-four chordal rings, FCCN-NET, NSFNET, ARPANET, European Optical Network (EON), and mesh-torus, for JET; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

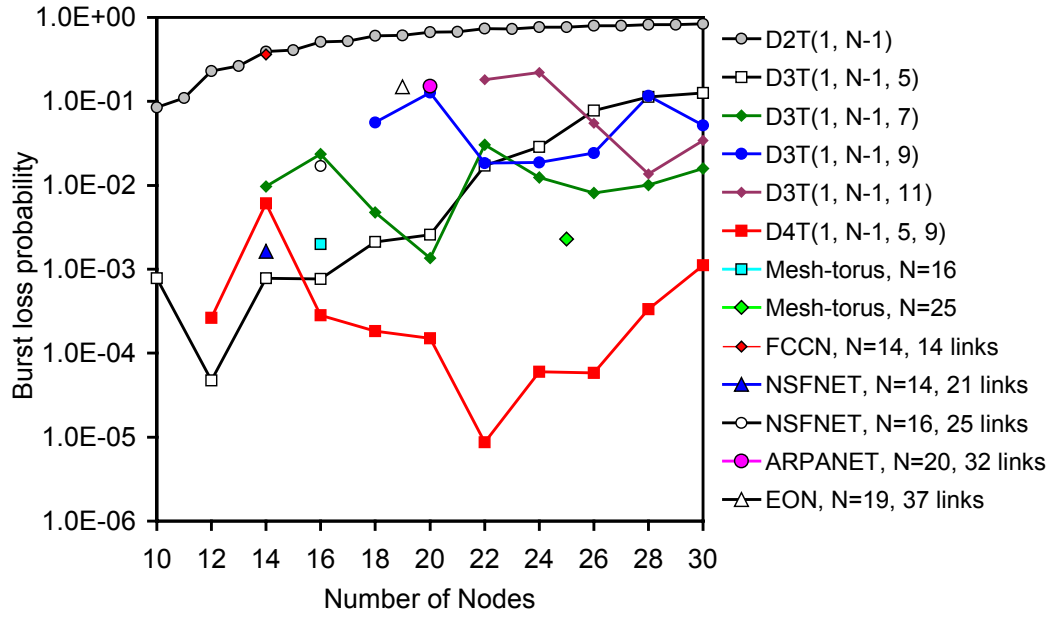


Fig. 5.16. Burst loss probability, as a function of the number of nodes (N), in the last hop of rings, degree-three, and degree-four chordal rings, FCCN-NET, NSFNET, ARPANET, European Optical Network (EON), and mesh-torus, for Horizon; F=64; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

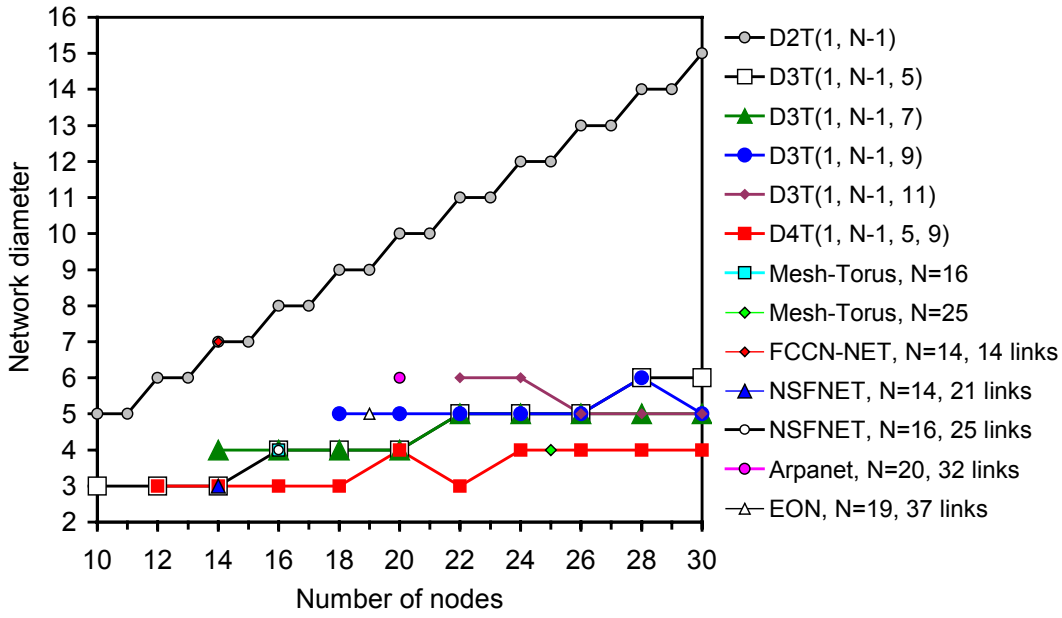


Fig. 5.17. Network diameter, as a function of the number of nodes (N), for rings, degree-three and degree-four chordal rings, FCCN-NET, NSFNET-NET, ARPANET, European Optical Network (EON), and mesh-torus networks.

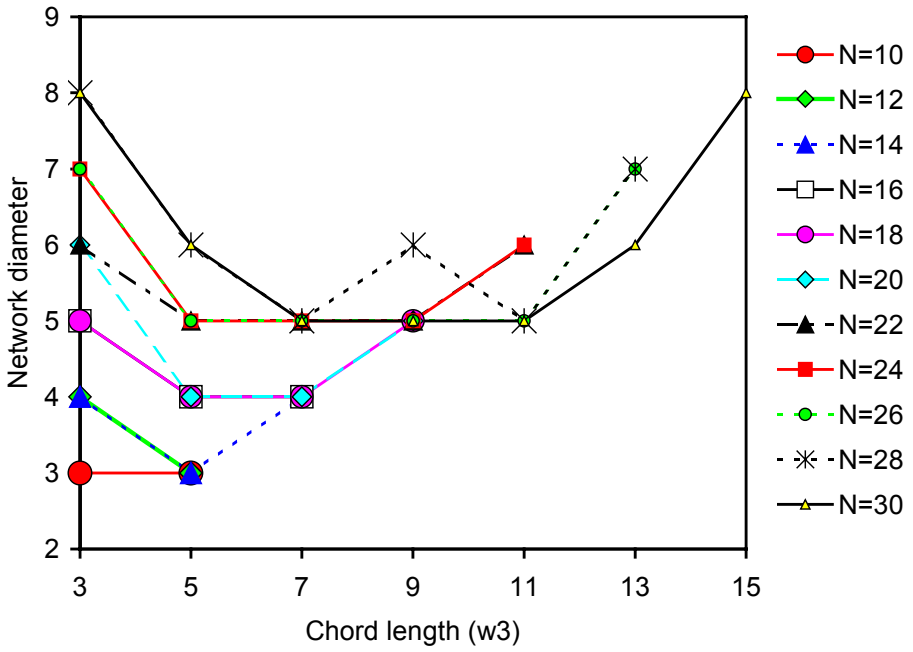


Fig. 5.18. Network diameter for degree-three chordal ring networks, as a function of the chord length (*w*3).

5.3 The Role of Nodal Degree in OBS Mesh Networks

This section focuses on performance assessment of one-way resource reservation protocols in OBS mesh networks. The performance analysis considers the five protocols under study and focuses on the role of nodal degree in OBS mesh networks. The nodal degree gain for FCCN-NET is not shown because it has the same nodal degree (equal to 2) as the correspondent ring (with *N*=14). Then, network topologies with nodal degree great than two are considered.

Firstly, network topologies with *N*=16 nodes are studied considering the following network topologies: NSFNET, Mesh-torus, and the corresponding ring and chordal rings between nodal degree-three and degree-six. Second, the study focuses on network topologies with around *N*=20 nodes, i.e., ARPANET, European Optical Network (EON) (with *N*=19), and the correspondent ring and chordal rings between nodal degree-three and degree-six for 20 nodes. Finally, the performance analysis focuses on mesh-topologies with nodal degree between two and six that are the following: rings, degree-three chordal rings, degree-four chordal rings, degree-five chordal rings, degree-six chordal rings, mesh-torus, NSFNET, ARPANET and EON.

5.3.1 Influence of Nodal Degree on the Performance of Networks with 16 Nodes

This sub-section discusses the performance implications of the nodal degree in OBS networks with mesh topologies and *N*=16 nodes.

Figures 5.19, 5.20, 5.21, 5.22, and 5.23 show, respectively for JIT, JumpStart, JIT$^+$, JET, and Horizon, the nodal degree gain, in the last hop of each topology, due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,5)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11). Concerning chordal rings, one has chosen among several topologies with smallest diameter the ones that led to the best network performance. As may be seen in those figures, the considered topologies may be sorted from the best performance to the worst performance as: D6T(1,15,3,5,7,11), D5T(1,15,7,3,9), D4T(1,15,5,13), D3T(1, 15, w_3), mesh-torus, and NSFNET. The performance of D5T(1,15,7,3,9) is close to the performance of D6T(1,15,3,5,7,11). These results are confirmed in Figures 5.24, 5.25, 5.26, 5.27, and 5.28 that display the nodal degree gain as a function of λ/μ, for the same topologies. In these figures it is possible to observe a similar behavior of the chordal rings for the five protocols under study. For values of λ/μ less than 32, the burst loss probability is equal to zero for all topologies, except for NSFNET that is zero for values of λ/μ less than 25.6.

Results presented in these figures (5.19 to 5.23) were obtained for the JIT, JumpStart, JIT$^+$, JET, and Horizon protocols, and, as may be seen, their performance is very close. For 16 nodes and 64 data channels per link, when the nodal degree gain due to the increase of nodal degree from 2 (rings) to 4, 5, and 6 (chordal ring with smallest diameter), a larger gain between four and five orders of magnitude is observed in the last hop of each topology. Concerning the nodal degree gain due to the increase of nodal degree from 2 (rings) to 3 (chordal ring with smallest diameter), it is about three orders of magnitude in the last hop of each topology for the same network conditions (*N*=16 and *F*=64). Comparing the nodal degree gain in the chordal-ring topologies, the largest difference between two consecutives nodal degrees is observed between nodal degree-three and four, being more than one order of magnitude, except for JIT$^+$ that is less than one order of magnitude. The performance of the five resource reservation protocols under study is very close.

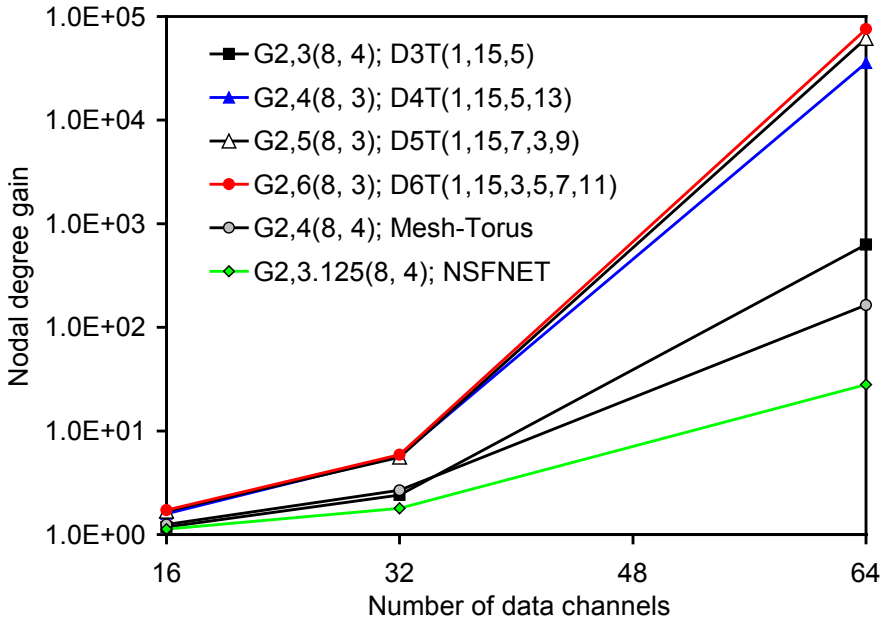


Fig. 5.19. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11) as a function of the number of data channels (F), in the last hop of each topology, for JIT; N=16; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

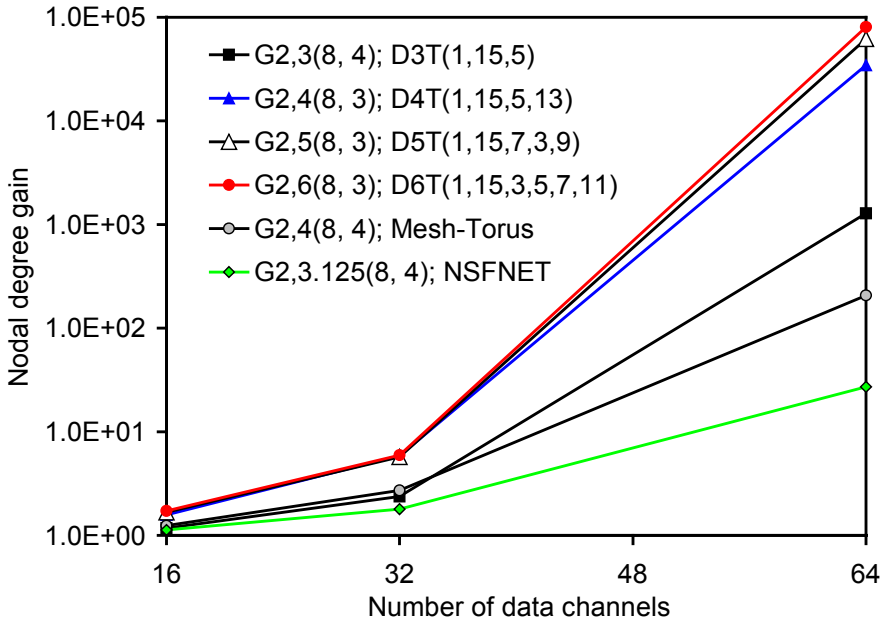


Fig. 5.20. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11), as a function of the number of data channels (F), in the last hop of each topology, for JumpStart; N=16; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

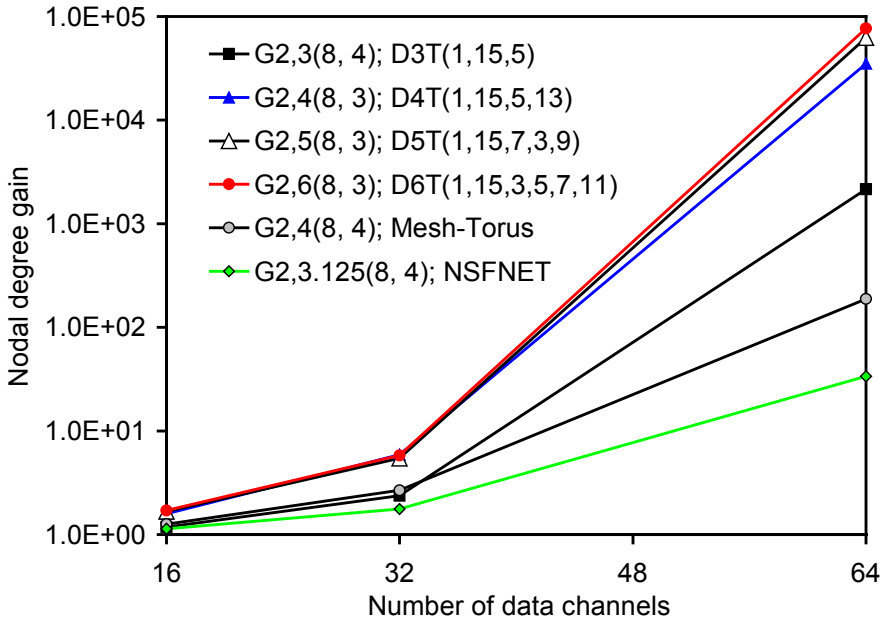


Fig. 5.21. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11) as a function of the number of data channels (F), in the last hop of each topology, for JIT$^+$; N=16; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

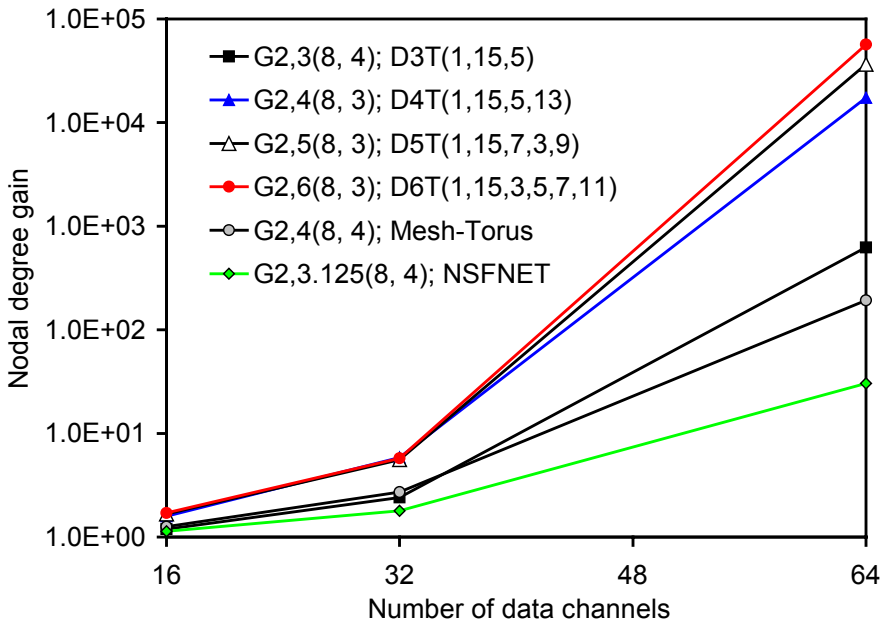


Fig. 5.22. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11) as a function of the number of data channels (F), in the last hop of each topology, for JET; N=16; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

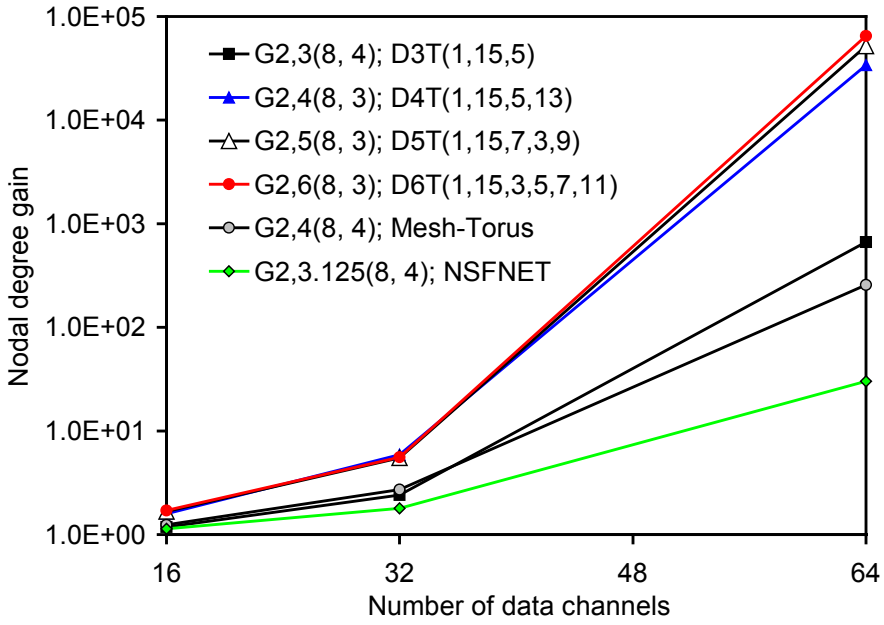


Fig. 5.23. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11) as a function of the number of data channels (F), in the last hop of each topology, for Horizon; N=16; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

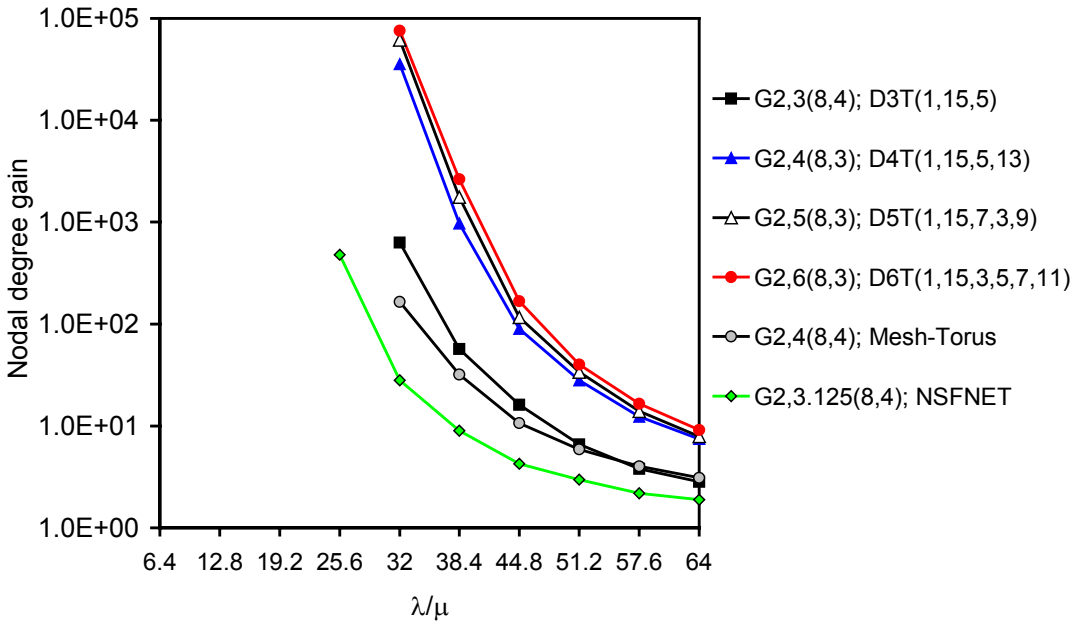


Fig. 5.24. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11), as a function of λ/μ, in the last hop of each topology, for JIT; N=16; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

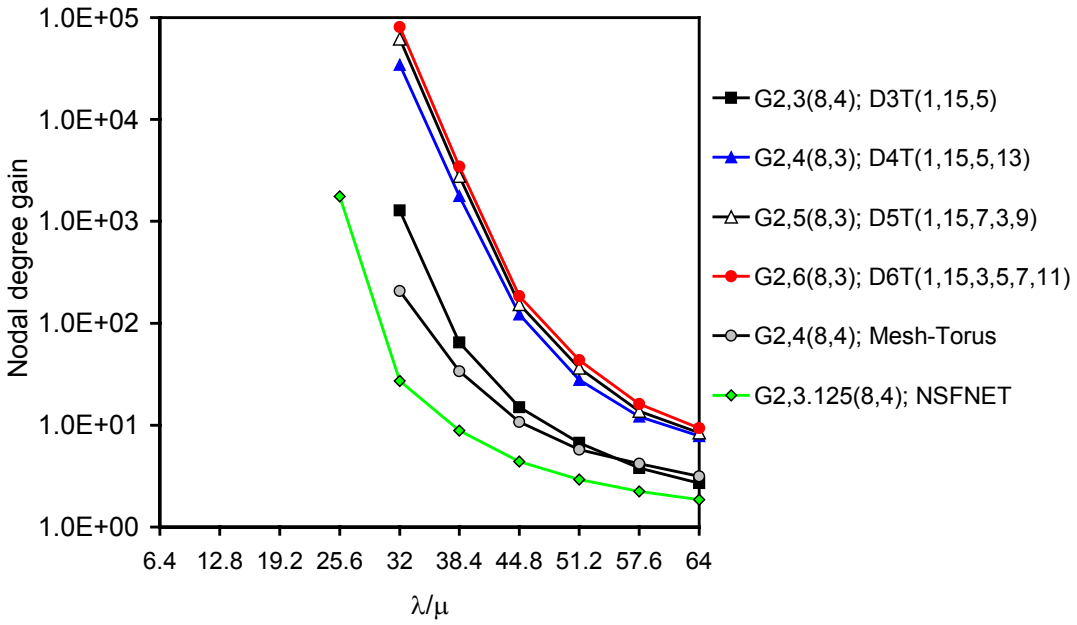


Fig. 5.25. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11), as a function of λ/μ, in the last hop of each topology, for JumpStart; N=16; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

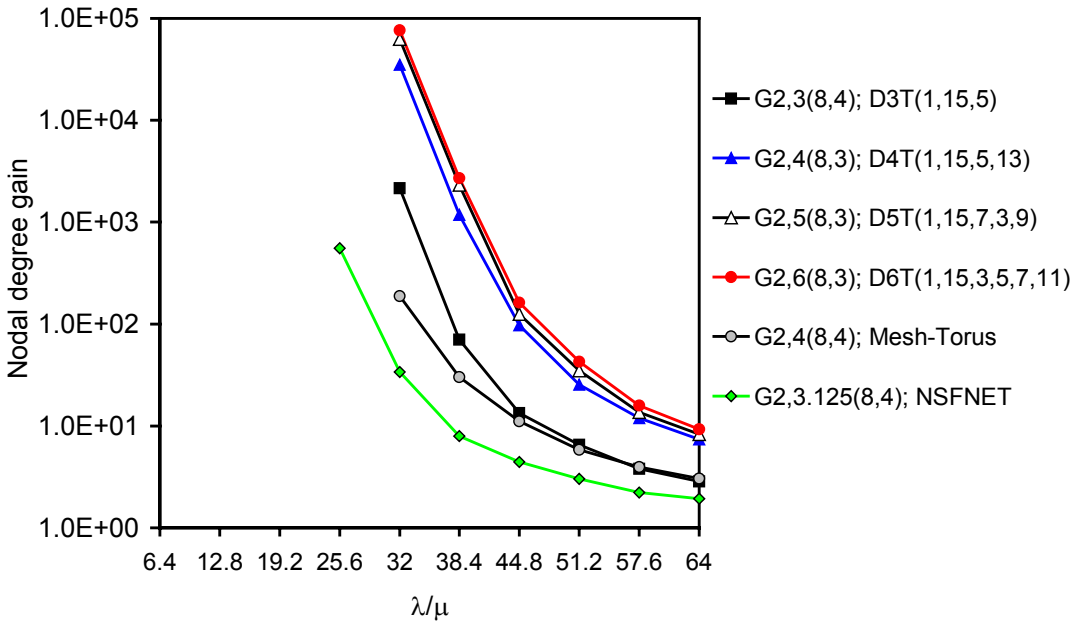


Fig. 5.26. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11), as a function of λ/μ, in the last hop of each topology, for JIT$^+$; N=16; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

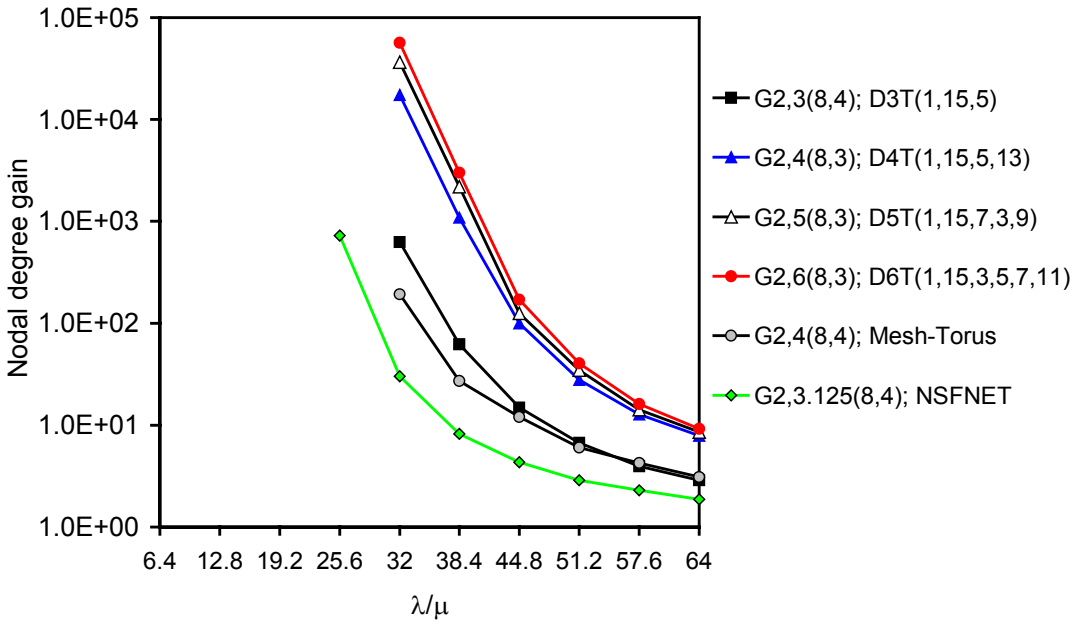


Fig. 5.27. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11), as a function of λ/μ, in the last hop of each topology, for JET; N=16; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

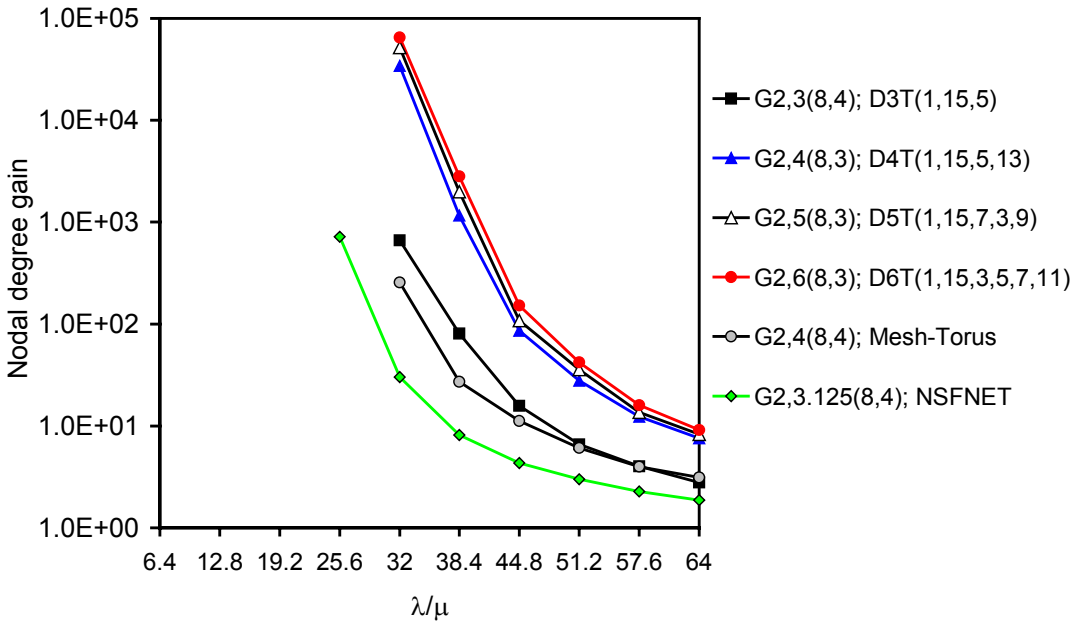


Fig. 5.28. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,15)) to: 3 (D3T(1,15,w_3)), 3.125 (NSFNET), 4 (D4T(1,15,5,13) and mesh-torus), 5 (D5T(1,15,7,3,9)), and 6 (D6T(1,15,3,5,7,11), as a function of λ/μ, in the last hop of each topology, for Horizon; N=16; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

5.3.2 Impact of Nodal Degree on the Performance of Networks with 20 Nodes

This sub-section is devoted to evaluate the performance implications of the nodal degree for OBS networks with mesh topologies and *N*=20 nodes.

Figures 5.29, 5.30, 5.31, 5.32, and 5.33 show, respectively for JIT, JumpStart, JIT$^+$, JET, and Horizon, the nodal degree gain, in the last hop of each topology, due to the increase of the nodal degree from 2 (D2T(1,19)) to: 3 (D3T(1,19,7)), 3.2 (ARPANET), 4 (D4T(1,19,3,9)), 5 (D5T(1,19,3,7,11)), and 6 (D6T(1,19,3,5,11,15)), and from 2 (D2T(1,18)) to 3.89 (EON – European Optical Network). Concerning chordal rings, the ones that led to the best network performance were chosen among several topologies with smallest diameter. As may be seen in those figures, the considered topologies may be sorted from the best performance to the worst performance as: D6T(1,19,3,5,11,15), D5T(1,19,3,7,11), D4T(1,19,3,9), D3T(1,19,7), ARPANET, and EON – European Optical Network. For networks with 20 nodes, when the nodal degree increases from 2 to 4 (chordal-rind), 5 and 6, the gain is between four and five orders of magnitude.

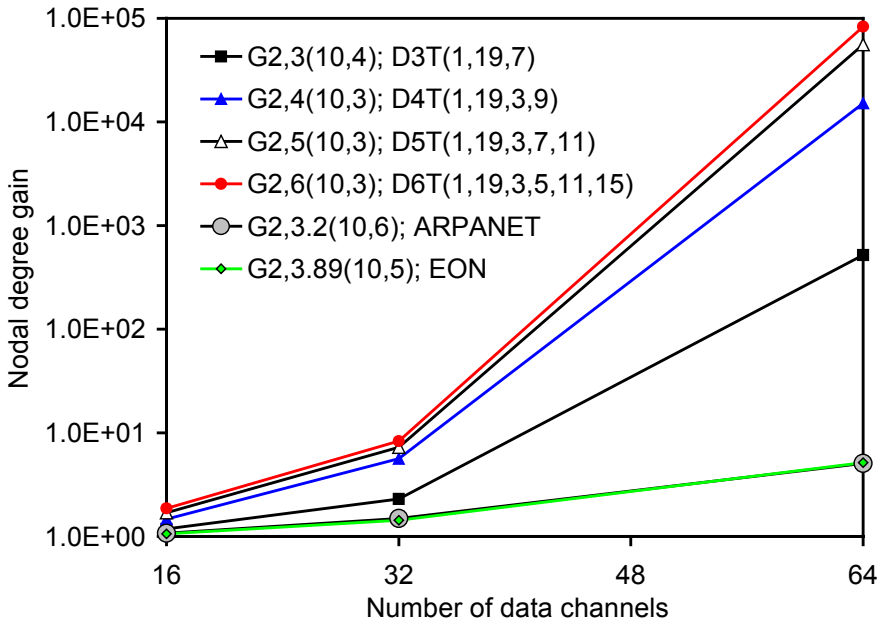


Fig. 5.29. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,19)) to: 3 (D3T(1,19,7)), 3.2 (ARPANET), 3.89 (EON), 4 (D4T(1,19,3,9)), 5 (D5T(1,19,3,7,11)), and 6 (D6T(1,19,3,5,11,15)) as a function of the number of data channels, in the last hop of each topology, for JIT; *N*=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

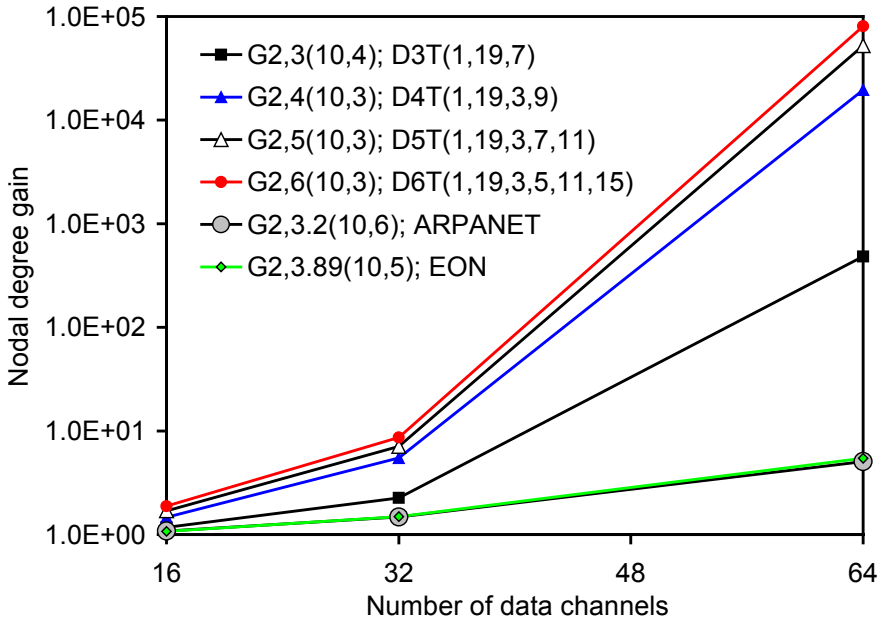


Fig. 5.30. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,19)) to: 3 (D3T(1,19,7)), 3.2 (ARPANET), 3.89 (EON), 4 (D4T(1,19,3,9)), 5 (D5T(1,19,3,7,11)), and 6 (D6T(1,19,3,5,11,15)) as a function of the number of data channels, in the last hop of each topology, for JumpStart; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

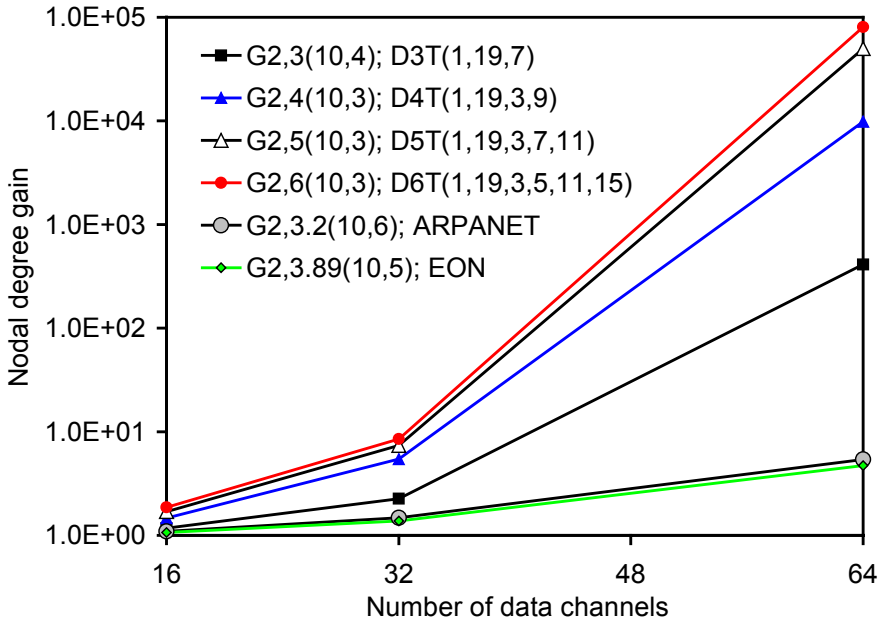


Fig. 5.31. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,19)) to: 3 (D3T(1,19,7)), 3.2 (ARPANET), 3.89 (EON), 4 (D4T(1,19,3,9)), 5 (D5T(1,19,3,7,11)), and 6 (D6T(1,19,3,5,11,15)) as a function of the number of data channels, in the last hop of each topology, for JIT$^+$; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

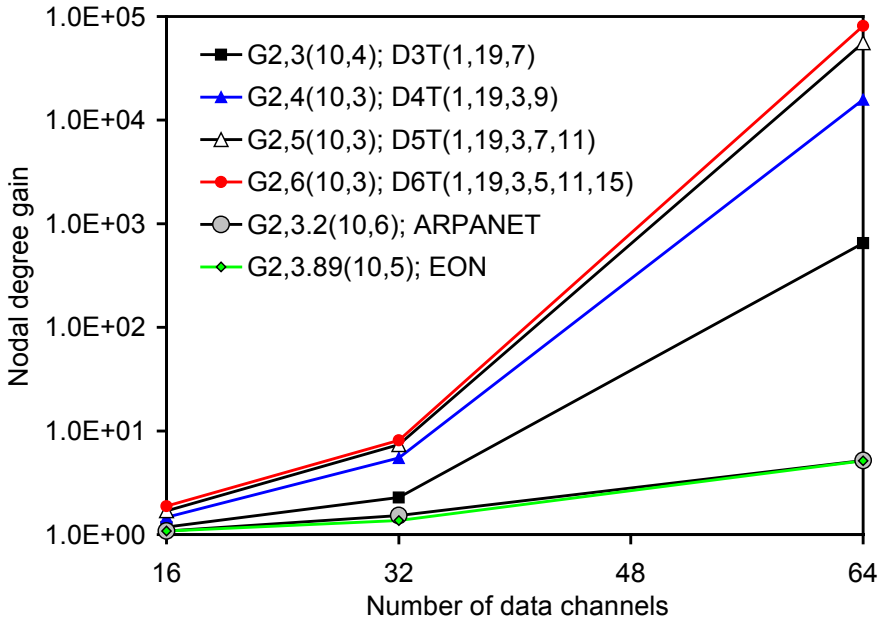


Fig. 5.32. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,19)) to: 3 (D3T(1,19,7)), 3.2 (ARPANET), 3.89 (EON), 4 (D4T(1,19,3,9)), 5 (D5T(1,19,3,7,11)), and 6 (D6T(1,19,3,5,11,15)) as a function of the number of data channels, in the last hop of each topology, for JET; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

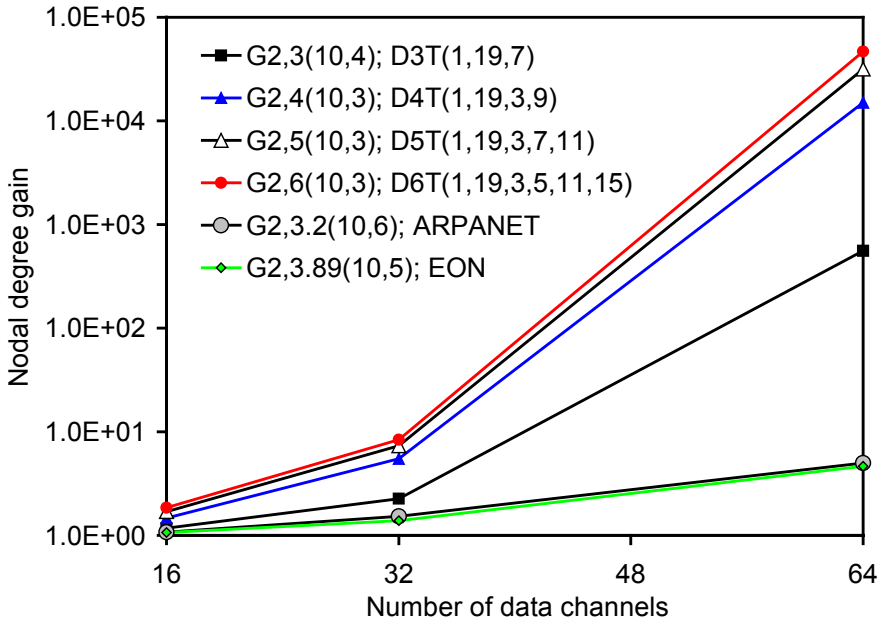


Fig. 5.33. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,19)) to: 3 (D3T(1,19,7)), 3.2 (ARPANET), 3.89 (EON), 4 (D4T(1,19,3,9)), 5 (D5T(1,19,3,7,11)), and 6 (D6T(1,19,3,5,11,15)) as a function of the number of data channels, in the last hop of each topology, for Horizon; N=20; λ/μ=32; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

It is observed that the performance of the ARPANET is very close to the performance of EON. ARPANET has a nodal degree (3.2) near to the degree-three topology (D3T(1,19,7)). However, although EON has a nodal degree (3.89) near to the degree-four topology (D4T(1,19,3,9)), the performance of degree-four topology is around four orders of magnitude better than EON. The performance of both ARPANET and EON is worst than the nearest chordal ring degree topology, being less than one order of magnitude. These results reveal the importance of the way links are connected in the network, since chordal rings and ARPANET and EON have similar nodal degrees and therefore a similar number of network links. Results presented in these figures (from 5.29 to 5.33) were obtained for the JIT, JumpSart, JIT$^+$, JET, and Horizon resource reservation protocols, and, as may be seen, their performance is very close. This result is confirmed in Figure 5.34, that presents the performance comparison of the nodal degree gain for the best (D6T(1,19,3,5,11,15)) and the worst (EON) of the topologies showed in Figures 5.29 to 5.33. Figure 5.34 shows the nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,18)) to 3.89 (EON), and 2 (D2T(1,19)) to 6 (D6T(1,19,3,5,11,15)), as a function of λ/μ, in the last hop of each topology, for JIT, JumpStart, JIT$^+$, JET, and Horizon protocols (F=64).

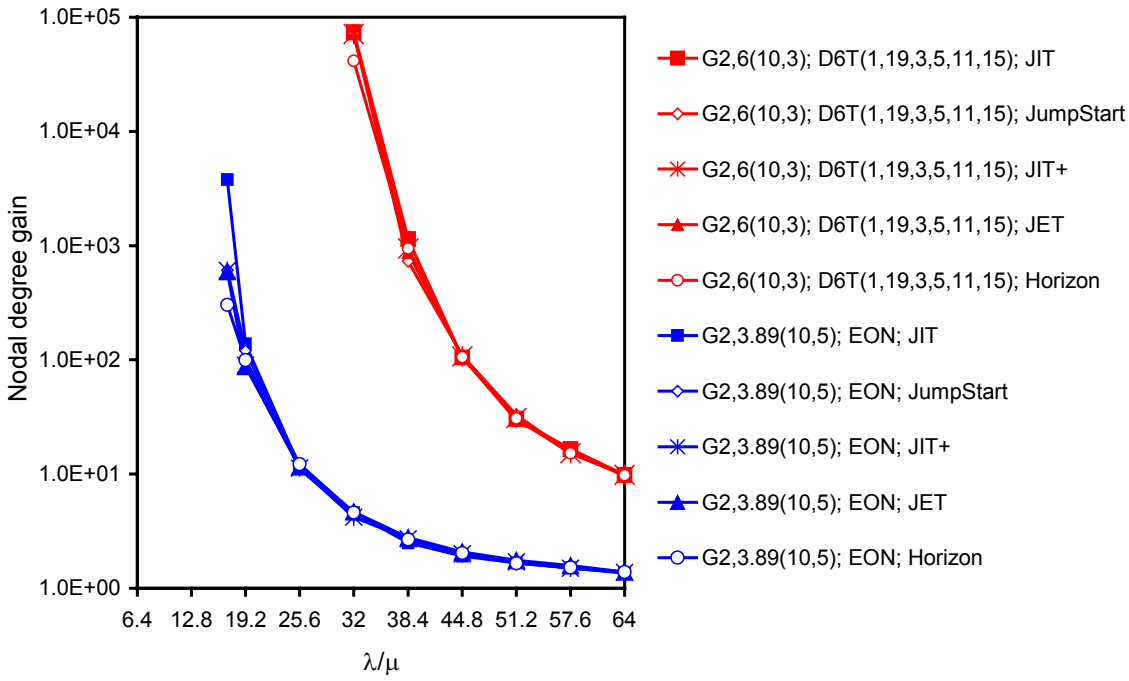


Fig. 5.34. Nodal degree gain due to the increase of the nodal degree from 2 (D2T(1,19)) to: 3.89 (EON), and 6 (D6T(1,19,3,5,11,15)), as a function of λ/μ, in the last hop of each topology, for JIT, JumpStart, JIT$^+$, JET, and Horizon protocols; N=20, F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

5.3.3 Influence of Nodal Degree on Nodal Degree Gain

This sub-section focuses on the study of the performance of OBS networks, based on the nodal degree gain, for the one-way resource reservation protocols considered. The performance analysis considers mesh topologies with nodal degree between three and six.

Figures 5.35, 5.36, 5.37, 5.38, and 5.39 plot, respectively for JIT, JumpStart, JIT$^+$, JET, and Horizon, the nodal degree gain in the last hop of each topology, as a function of the nodal degree, due to the increase of the nodal degree from 2 (D2T(1,13)) to 3 (NSFNET (*N*=14)), from 2 (D2T(1,15)) to: 3 (D3T(1,15,5)), 3.125 (NSFNET (*N*=16)), 4 (D4T(1,15,5,13) and Mesh-Torus (*N*=16)), 5 (D5T(1,15,7,3,9), and 6 (D6T(1,15,3,5,7,11)), from 2 (D2T(1,18)) to 3.89 (EON (*N*=19)), from 2 (D2T(1,19)) to: 3 (D3T(1,19,7)), 3.2 (ARPANET (*N*=20)), 4 (D4T(1,19,3,9)), 5 (D5T(1,19,3,7,11)), 6 (D6T(1,19,3,5,11,15)), from 2 (D2T(1,24)) to 4 (Mesh-Torus (*N*=25)), and from 2 (D2T(1,29)) to 6 (D6T(1,29,3,7,11,13)) (*F*=64). As may be seen, when the nodal degree increases from 2 to around 3, the largest gain is observed for degree-three chordal rings (slightly less than three orders of magnitude) and the smallest gain is observed for the ARPANET (less than one order of magnitude). When the nodal degree increases from 2 to around 4, the largest gain is observed for degree-four chordal rings (with a gain between four and five orders of magnitude) and the smallest gain is observed for the EON (with a gain less than one order of magnitude). When the nodal degree increases from 2 to around 5 or 6, the gain is between four and six orders of magnitude depending on the number of nodes. These results clearly show the importance of the way links are connected in OBS networks, since, in this kind of networks, burst loss probability is a key issue.

In Figure 5.40, the nodal degree gain in the last hop of each topology is compared, as a function of the nodal degree, due to the increase of the nodal degree from 2 (D2T(1,15)) to 3 (D3T(1,15,5)) and 4 (D4T(1,15,5,13)), and from 2 (D2T(1,19)) to 5 (D5T(1,19,3,7,11)), for JIT, JumpStart, JIT$^+$, JET, and Horizon resource reservation protocols (*F*=64). Topologies leading to the best performances for nodal degree of 3, 4, and 5 have been considered. This figure confirms previous results where the performance of the resource reservation protocols considered is very close.

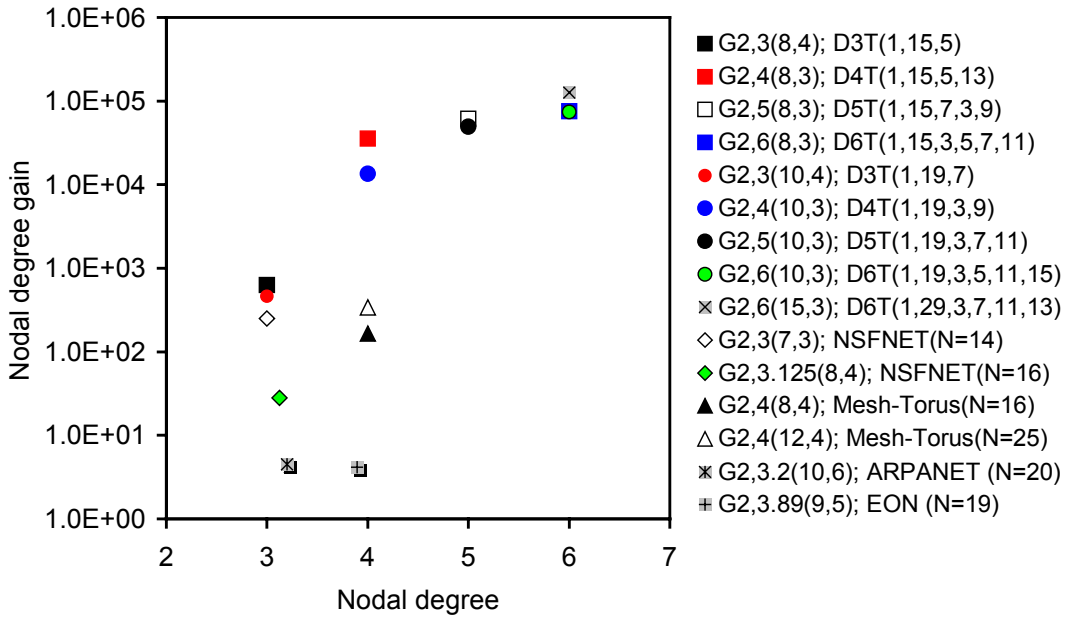


Fig. 5.35. Nodal degree gain in the last hop of each topology, as a function of the nodal degree for JIT resource reservation protocol; λ/μ=32; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

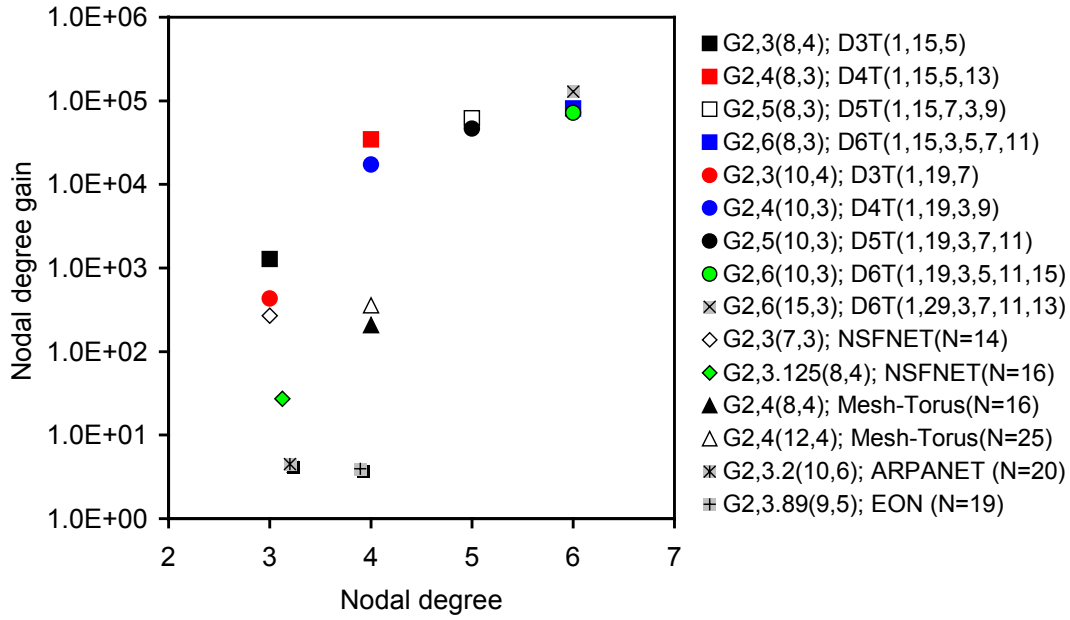


Fig. 5.36. Nodal degree gain in the last hop of each topology, as a function of the nodal degree for JumpStart resource reservation protocol; λ/μ=32; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

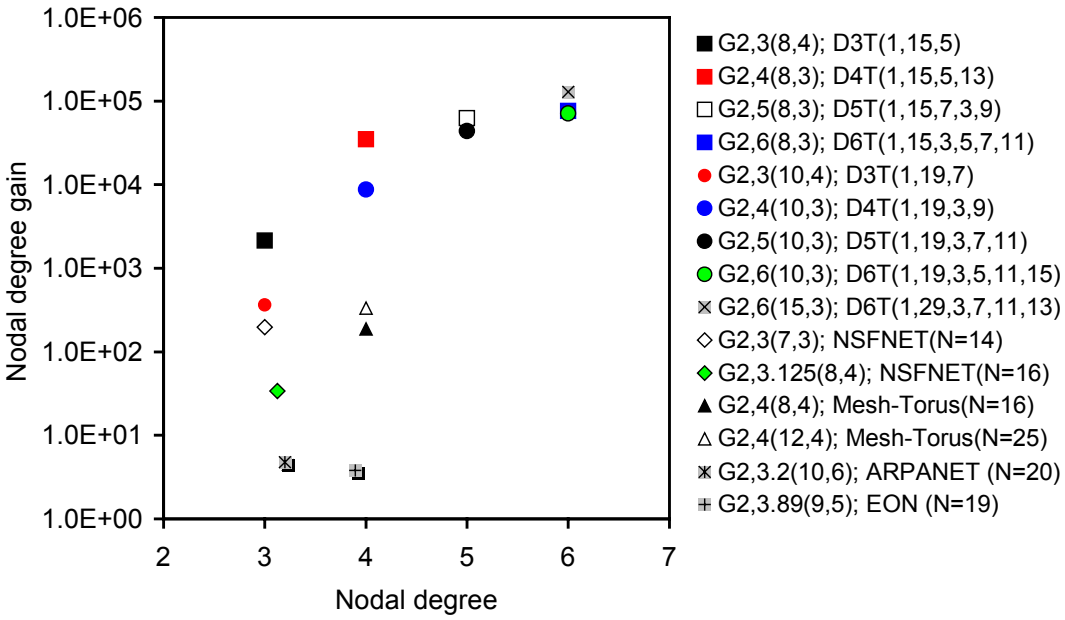


Fig. 5.37. Nodal degree gain in the last hop of each topology, as a function of the nodal degree for JIT$^+$ resource reservation protocol; λ/μ=32; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

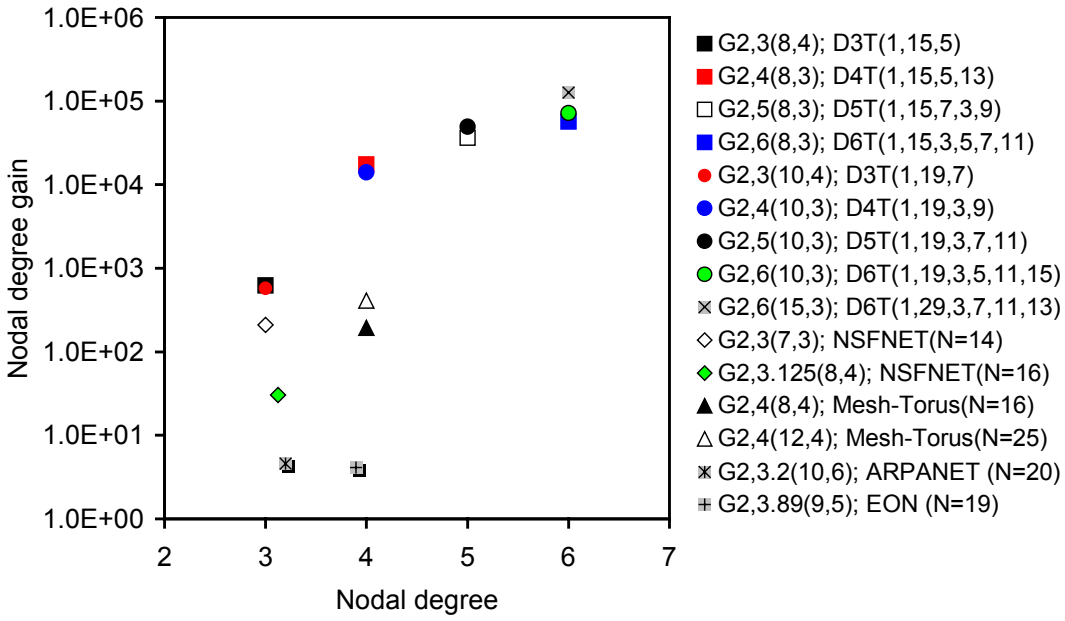


Fig. 5.38. Nodal degree gain in the last hop of each topology, as a function of the nodal degree for JET resource reservation protocol; λ/μ=32; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

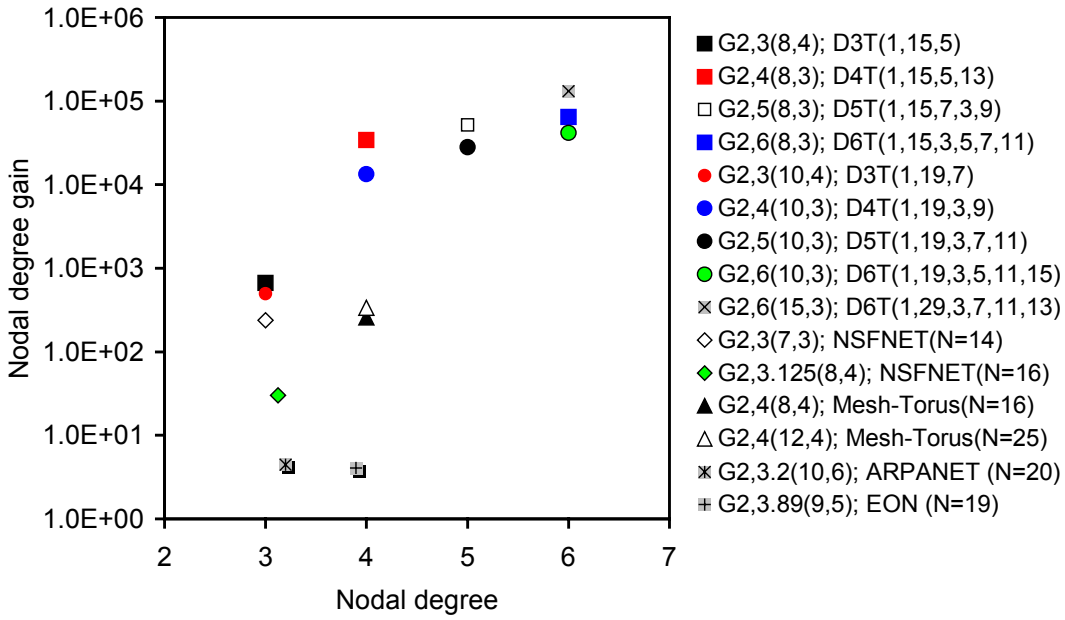


Fig. 5.39. Nodal degree gain in the last hop of each topology, as a function of the nodal degree for Horizon resource reservation protocol; λ/μ=32; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

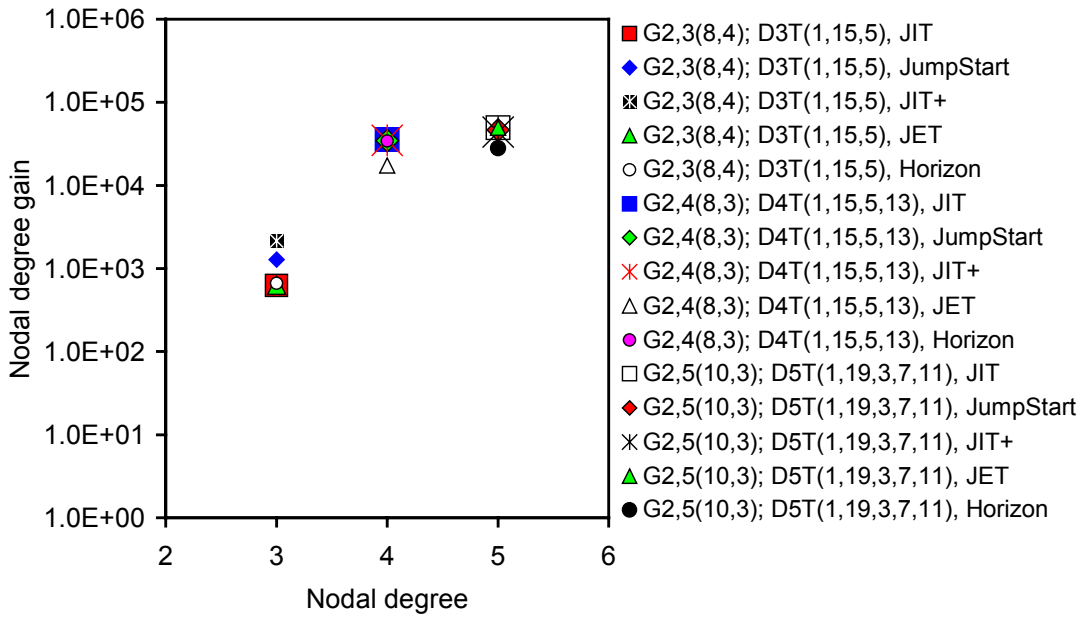


Fig. 5.40. Nodal degree gain in the last hop of each topology, as a function of the nodal degree for JIT, JumpStart, JIT$^+$, JET, Horizon resource reservation protocols; λ/μ=32; F=64; T_{OXC}=10ms; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

5.4 Impact of Setup Message Processing Time and OXC Configuration Time

This section studies the influence of setup message processing time and OXC configuration time on the performance of OBS mesh networks for JIT, JumpStart, JIT$^+$, JET, and Horizon protocols. Network topologies are grouped by nodal degree considering the network topologies with nodal degree around three and nodal degree around four. NSFNET with N=14 nodes has nodal degree equal to 3, NSFNET with N=16 nodes has nodal degree equal to 3.125, and ARPANET with nodal degree equal to 3.2 are examples of nodal degree around three. Mesh-torus with N=16 and N=25 have nodal degree equal to 4, and European Optical Network (EON) has nodal degree equal to 3.89 are examples of nodal-degree around four. Chordal ring networks with smallest diameter, which present best performance, selected for degree-three and degree-four network topologies are D3T(1,19,7) and D4T(1,19,3,9), respectively.

Figure 5.41 shows burst loss probability as a function of OXC configuration time in the last hop of NSFNET (with N=14 and N=16), ARPANET, and D3T(1,19,7) for the five protocols under study, with F=64 and λ/μ=32. In this figure a fixed value for T_{Setup} time is assumed, which is the value defined for JIT, JumpStart, and JIT$^+$ and estimated for JET and Horizon for currently available technology [80]. T_{OXC} is assumed to range from the value estimated for a near future scenario (T_{OXC}=20μs) up to ten times the value defined for currently available technology, i.e. T_{OXC}=10*10ms=100ms. As may be seen in this figure, chordal rings and NSFNET with N=14 nodes clearly have better performance than ARPANET and NSFNET with N=16 nodes for $T_{OXC}\leq$50ms. It may also be observed that for $T_{OXC}\leq$1ms, the performance of these networks is independent of the change of the T_{OXC}, which means that a reduction of the values of T_{OXC} to ones smaller than 1ms does not improve the network performance. Moreover, it may also be observed that the relative performance of the five resource reservation protocols is similar, being JIT and JIT$^+$ slightly better than the other ones for chordal rings when $T_{OXC}\leq$1ms.

Figures showing the burst loss probability, as a function of setup message processing time for NSFNET (with N=14 and N=16), ARPANET, and D3T(1,19,7), and for mesh-torus (with N=16 and N=25), EON, and D4T(1,19,3,9), considered below, are not included because it was observed that the reduction of T_{Setup} does not lead to a better network performance. These results confirm previous observations presented in Figures 4.22 and 4.39 of Chapter 4.

Figure 5.42 confirms results found in Figure 5.41. Figure 5.42 illustrates the burst loss probability as a function of OXC configuration time in the last hop of NSFNET (with N=14 and N=16), ARPANET, and D3T(1,19,7) for the five protocols under study, with F=64 and λ/μ=32. T_{Setup} is assumed to change with T_{OXC}, according to (4.3), (4.4), and (4.5). As may be seen, for $T_{OXC}\leq$1ms, a small change in the burst loss can be observed regarding Figure 5.41. However, this small change is not significant in terms of network performance.

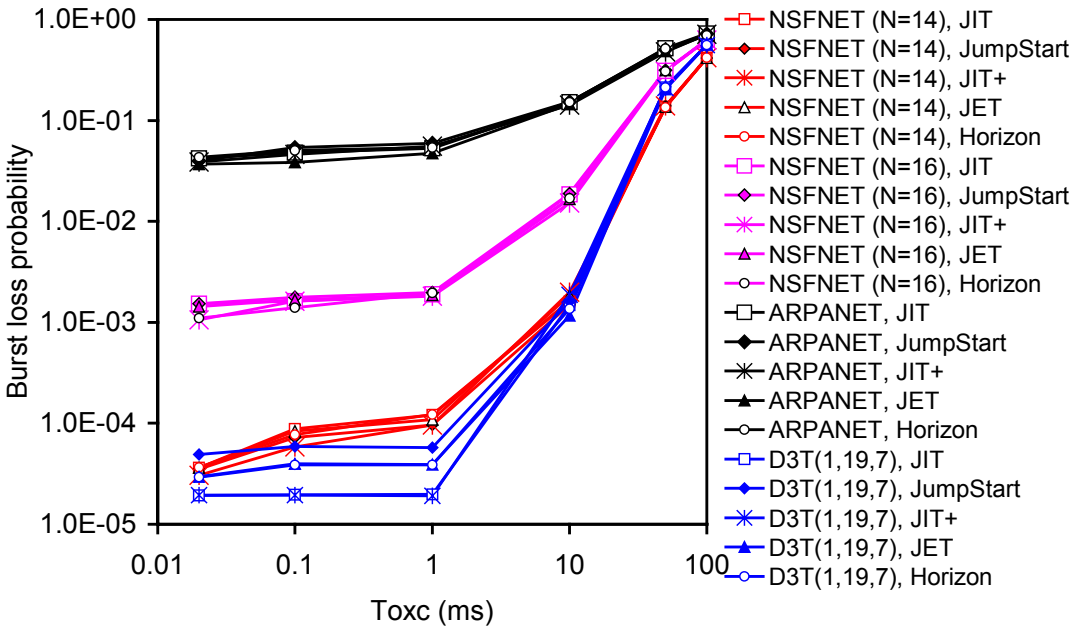


Fig. 5.41. Burst loss probability as a function of OXC configuration time in the last hop of NSFNET (N=14), NSFNET (N=16), ARPANET, and D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; λ/μ=32; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

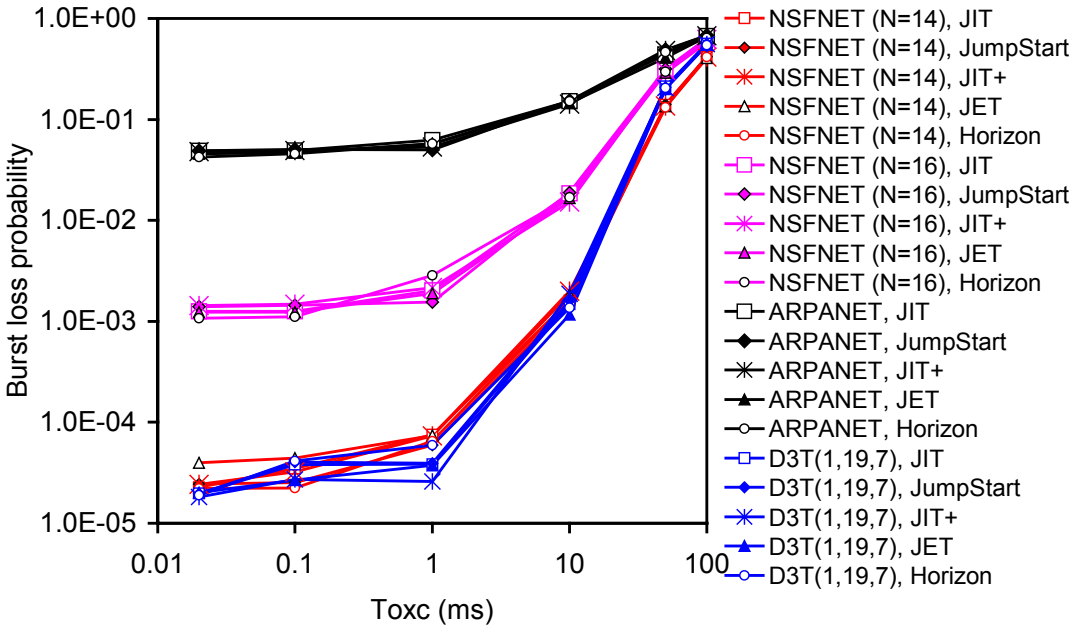


Fig. 5.42. Burst loss probability as a function of OXC configuration time in the last hop of NSFNET (N=14), NSFNET (N=16), ARPANET, and D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; λ/μ=32; with changing T_{Setup} according to (4.3), (4.4), and (4.5) for each resource reservation protocol.

Figure 5.43 plots burst loss probability as a function of setup message processing time in the last hop of the same network topologies shown above in Figures 5.41 and 5.42 for JIT, JumpStart, JIT$^+$, JET, and Horizon protocols, with F=64 and λ/μ=32. T_{Setup} assumes the values ranging between 1μs and 12.5μs, considering 5μs and 10μs as intermediate times, for JIT, JumpStart, and JIT$^+$ protocols. The value of T_{OXC} is given by (4.6). T_{Setup} for JET assumes the values ranging between 4μs and 50μs, considering 20μs and 30μs as intermediate times. T_{OXC} is given by (4.7). T_{Setup} for Horizon assumes the values ranging between 2μs and 25μs, considering 10μs and 20μs as intermediate times. T_{OXC} is given by (4.8). T_{OXC} is assumed to change with T_{Setup}, being computed according to (4.6), (4.7), and (4.8). As may be seen, chordal rings and NSFNET with N=14 nodes clearly have better performance than ARPANET and NSFNET with N=16. When the value of T_{Setup} increases, the correspondent burst loss probability also increases. For chordal rings and NSFNET with N=14 nodes, for values ranging between the first and the last times considered for each T_{Setup} and every protocol, the burst loss probability increases around two orders of magnitude. For NSFNET with N=16 nodes, the burst loss

probability increases around one order of magnitude, and for ARPANET the burst loss probability increases less than one order of magnitude. This figure also confirms previous results in terms of the best performance of chordal rings and NSFNET with N=14 nodes in comparison with NSFNET with N=16 nodes and ARPANET, and the influence of T_{OXC} and T_{Setup} in the network performance.

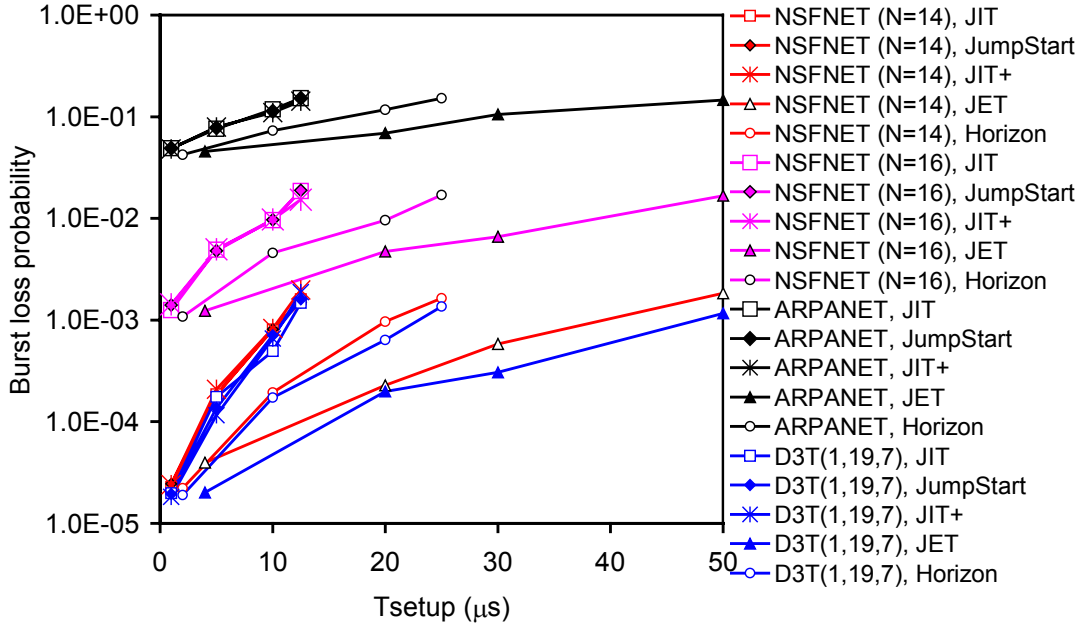


Fig. 5.43. Burst loss probability *versus* T_{Setup} in the last hop of NSFNET (N=14), NSFNET (N=16), ARPANET, and D3T(1,19,7) for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; λ/μ=32, with changing T_{OXC} according to (4.6), (4.7), and (4.8) for each protocol.

The next figures (Figures 5.44, 5.45, and 5.46) illustrate the same network conditions as previous Figures 5.41, 5.42, and 5.43 for network topologies with nodal degree around four, changing only the value of λ/μ to 44.8. Network topologies under study are mesh-torus (with N=16 and N=25), EON, and the degree-four chordal ring D4T(1,19,3,9). As may be seen in Figure 5.44, the degree-four chordal ring has the best performance and the worst is presented by EON. Mesh-torus presents a similar performance for both number of nodes considered. This result is explained by the same nodal degree (four) and the same way of connections between their nodes. For values of T_{OXC} less than 1ms, the performance of each network does not improve. This observation confirms previous results where the performance of the networks is independent of the change of the T_{OXC}. Additionally, it may also be observed that the relative performance of the five resource reservation protocols is similar. These results are confirmed in Figure 5.45.

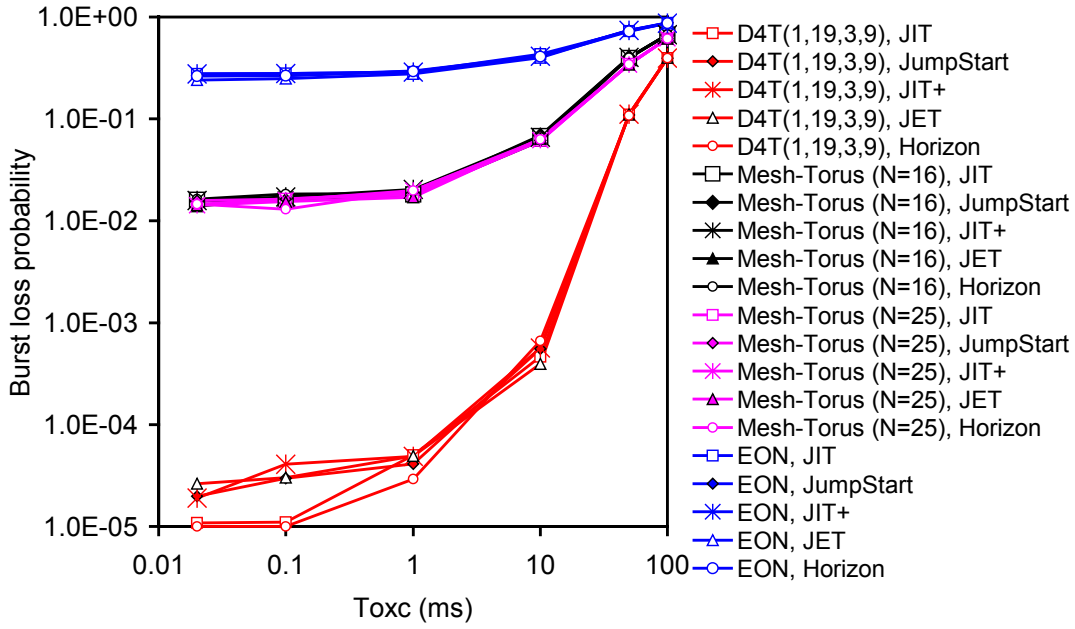


Fig. 5.44. Burst loss probability as a function of OXC configuration time in the last hop of D4T(1,19,3,9), Mesh-torus (N=16), Mesh-Torus (N=25), and EON for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; λ/μ=44.8; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

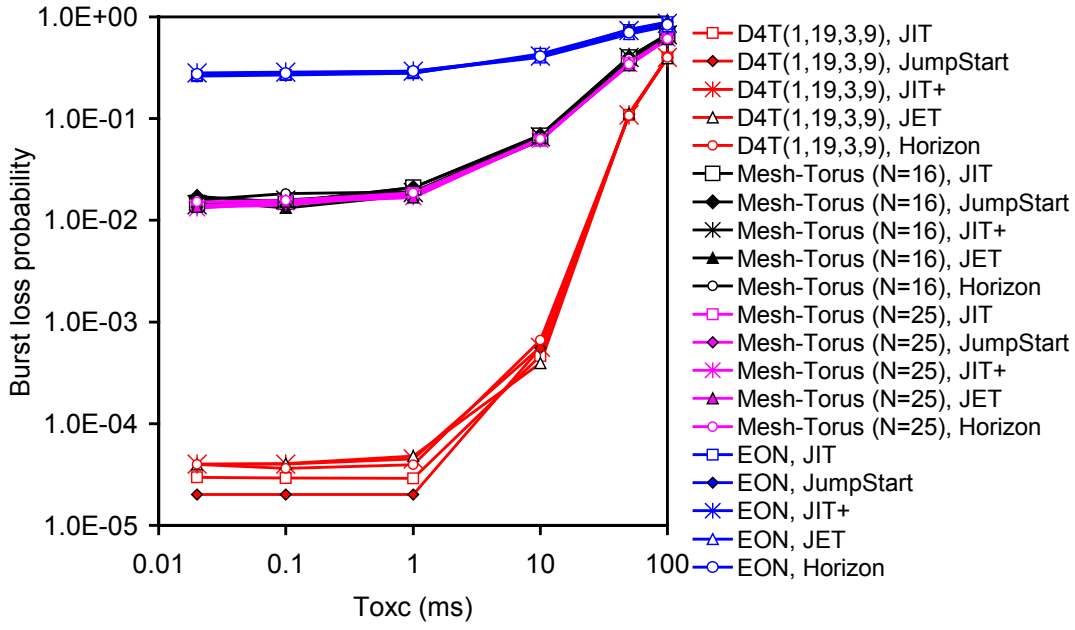


Fig. 5.45. Burst loss probability as a function of OXC configuration time in the last hop of D4T(1,19,3,9), Mesh-torus (N=16), Mesh-Torus (N=25), and EON for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; λ/μ=44.8; with changing T_{Setup} according to (4.3), (4.4), and (4.5) for each resource reservation protocol.

Figure 5.46 confirms previous results. As may be seen, the degree-four chordal ring performs better and, for values of T_{Setup} between the first and last value considered, the burst loss probability increases more than one order of magnitude. Mesh-torus has a similar performance and the worst performance is presented by EON. The performance of the five protocols is very similar.

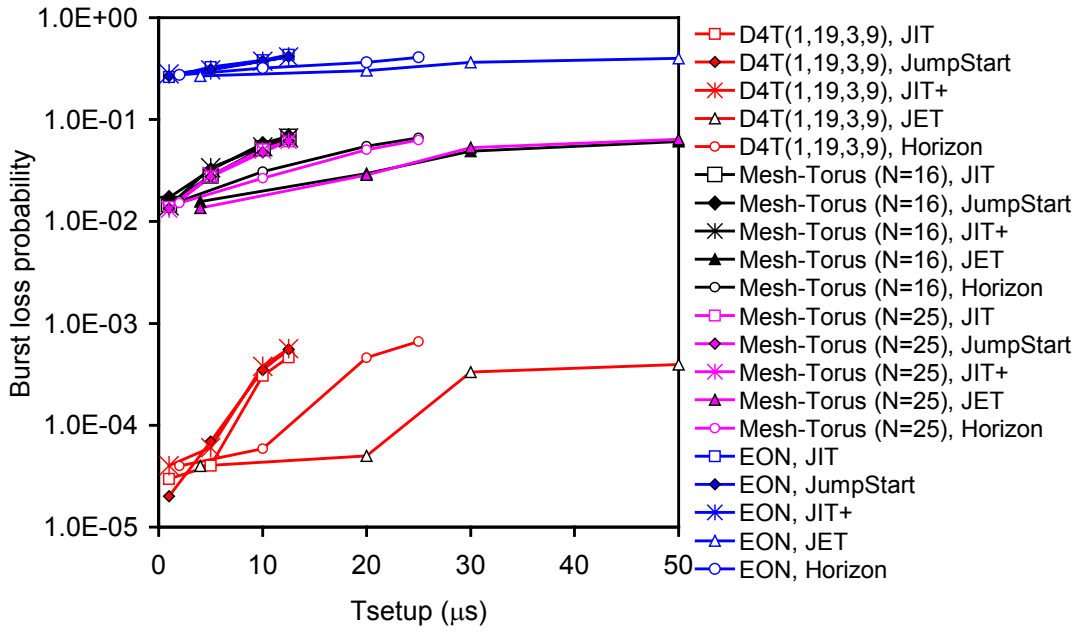


Fig. 5.46. Burst loss probability *versus* T_{Setup} in the last hop of D4T(1,19,3,9), Mesh-torus (N=16), Mesh-Torus (N=25), and EON for JIT, JumpStart, JIT$^+$, JET, and Horizon; F=64; λ/μ=44.8, with changing T_{OXC} according to (4.6), (4.7), and (4.8) for each resource reservation protocol.

5.5 Performance Assessment of E-JIT Resource Reservation Protocol in OBS Networks

E-JIT is an OBS one-way resource reservation protocol proposed in this thesis and it was described in Sub-section 2.4.3. Therefore, it is necessary to evaluate the performance of this new protocol in comparison with others. In above sections, it was observed that the performance of the five one-way resource reservation protocols under study is very similar (JIT, JumpStart, JIT$^+$, JET, and Horizon). On the other hand, E-JIT is a JIT based protocol. Therefore, this section studies the

performance of E-JIT protocol in OBS networks, in comparison with JIT. The performance evaluation is studied for rings, degree-three and degree-four chordal rings, FCCN-NET, NSFNET (with *N*=14 and *N*=16), Mesh-torus (with *N*=16 and *N*=25), ARPANET and EON network topologies.

5.5.1 Performance Assessment for Networks with 16 and 20 Nodes

This sub-section focuses on the study of performance assessment of OBS networks with mesh topologies with *N*=16 and *N*=20 nodes for JIT and E-JIT resource reservation protocols. D3T(1,15,5) and D3T(1,19,7) are the degree-three chordal ring networks with smallest diameter, which present best performance for *N*=16 and *N*=20 nodes, respectively. D4T(1,15,5,13) and D4T(1,19,3,9) are the best performance degree-four chordal ring networks with smallest diameter for *N*=16 and *N*=20 nodes, respectively.

Figure 5.47 plots burst loss probability as a function of number of data channels per link for JIT and E-JIT protocols, in the last hop of ring (D2T(1,15)), chordal ring (D3T(1,15,5) and D4T(1,15,5,13)), NSFNET, and Mesh-Torus networks with *N*=16 and λ/μ=32. As may be seen, when enough network resources are available (*F*=64), chordal rings with high nodal degree have the best performance and the ring presents the worst performance. Network topologies can be sorted from the best to the worst performance as: D4T(1,15,5,13), D3T(1,15,5), Mesh-Torus, NSFNET, and D2T(1,15). These results confirm previous observations in terms of relative performance of network topologies. However, the most important result of this and the next figures is the relative best performance of E-JIT in comparison with JIT. When the burst loss probability is smaller, the performance improvement of E-JIT over JIT is more significant.

Figure 5.48 confirms observations made for Figure 5.47. Figure 5.48 illustrates the burst loss probability as a function of number of hops for the same network topologies and conditions considered previously. In the first hop of D4T(1,15,5,13), the value of the burst loss probability tends to zero. This topology has better performance than others and it has the smallest network diameter (three). For network topologies with same network diameter, equal to four, degree-three chordal ring performs better, followed by Mesh-torus and NSFNET. The ring, with the largest network diameter equal to eight, performs worst than the other topologies.

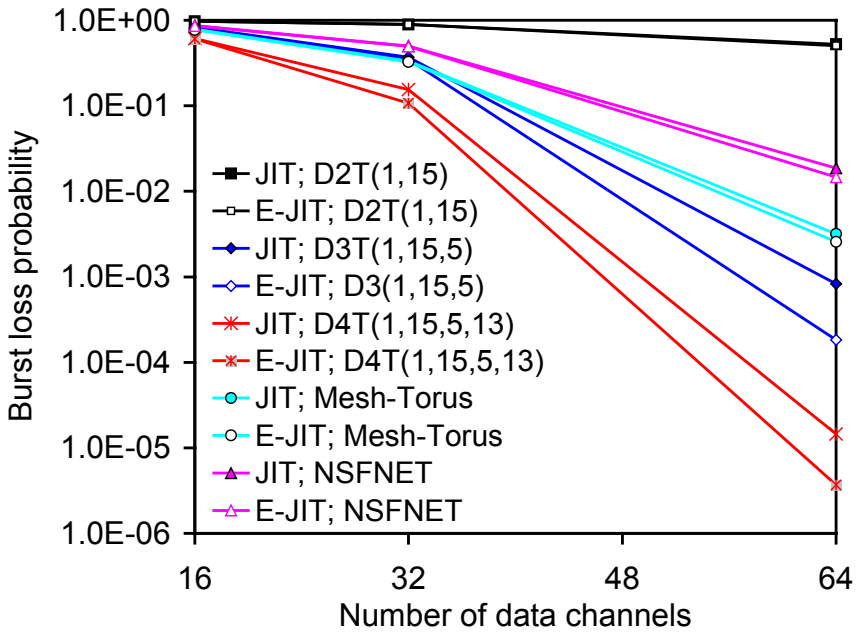


Fig. 5.47. Burst loss probability, as a function of number of number of data channels per link (*F*), in the last hop of ring (D2T(1,15)), chordal ring (D3T(1,15,5) and D4T(1,15,5,13)), NSFNET, and Mesh-Torus networks for JIT and E-JIT protocols; λ/μ=32; *N*=16; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

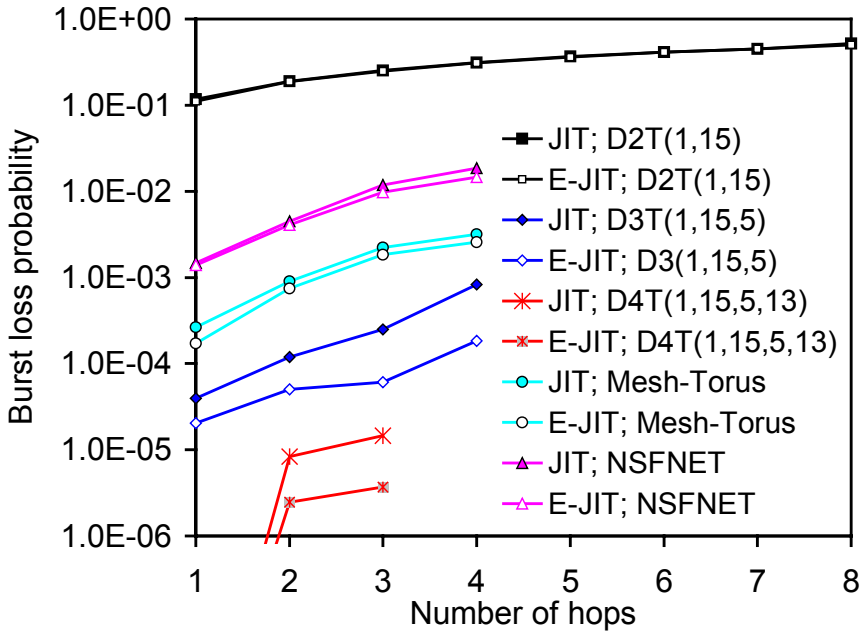


Fig. 5.48. Burst loss probability, as a function of number of number of hops, for ring (D2T(1,15)), chordal ring (D3T(1,15,5) and D4T(1,15,5,13)), NSFNET, and Mesh-Torus networks using JIT and E-JIT protocols; λ/μ=32; *F*=64; *N*=16; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

Previous figures considered topologies with N=16 nodes and the next figures present topologies around N=20 nodes. Figure 5.49 shows the burst loss probability as a function of number of data channels per link for JIT and E-JIT resource reservation protocols, in the last hop of ring (D2T(1,19)), chordal rings (D3T(1,19,7) and D4T(1,19,3,9)), ARPANET (N=20), and EON (N=19) with λ/μ=32. As may be seen, this figure confirms that chordal rings with high nodal degree have better performance than other topologies. It also confirms that E-JIT performs better than JIT, mainly, when the burst loss probability is minor. Another observation is related with the similar performance of ARPANET in comparison with EON. Figure 5.50 also confirms these results which plots the burst loss probability as a function of number of hops for ring (D2T(1,19)), chordal rings (D3T(1,19,7) and D4T(1,19,3,9)), ARPANET (N=20), and EON (N=19) for JIT and E-JIT resource reservation protocols, with λ/μ=32. Although the value of the EON network diameter is smaller than the network diameter of ARPANET, their performance is similar in the last hop of each one. This reveals the importance of how links are interconnecting nodes. For other topologies, networks with minor network diameter lead to better performance. In terms of performance of protocols under study, for degree-four chordal ring, E-JIT performs better than JIT around one order of magnitude.

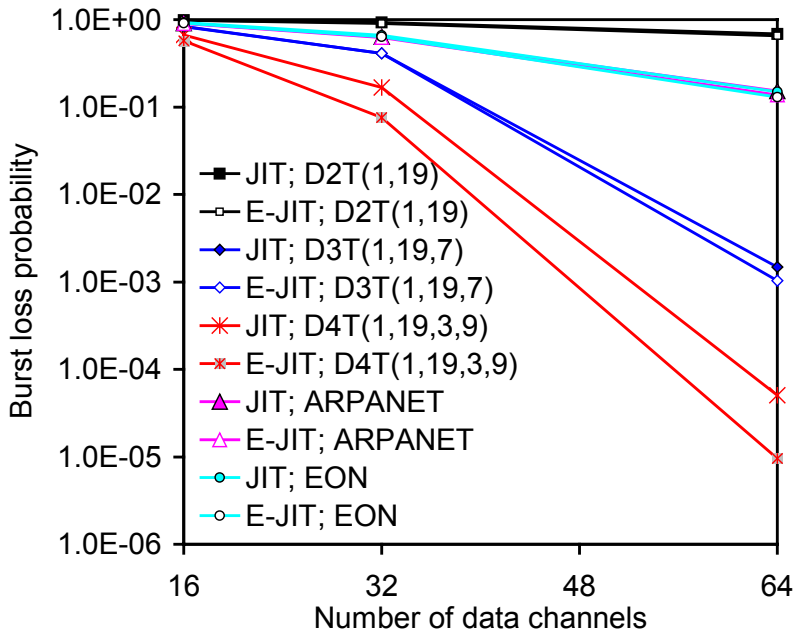


Fig. 5.49. Burst loss probability, as a function of number of number of data channels per link (F), in the last hop of ring (D2T(1,19)), chordal ring (D3T(1,19,7) and D4T(1,19,3,9)), ARPANET (N=20), and EON (N=19) for JIT and E-JIT protocols; λ/μ=32; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

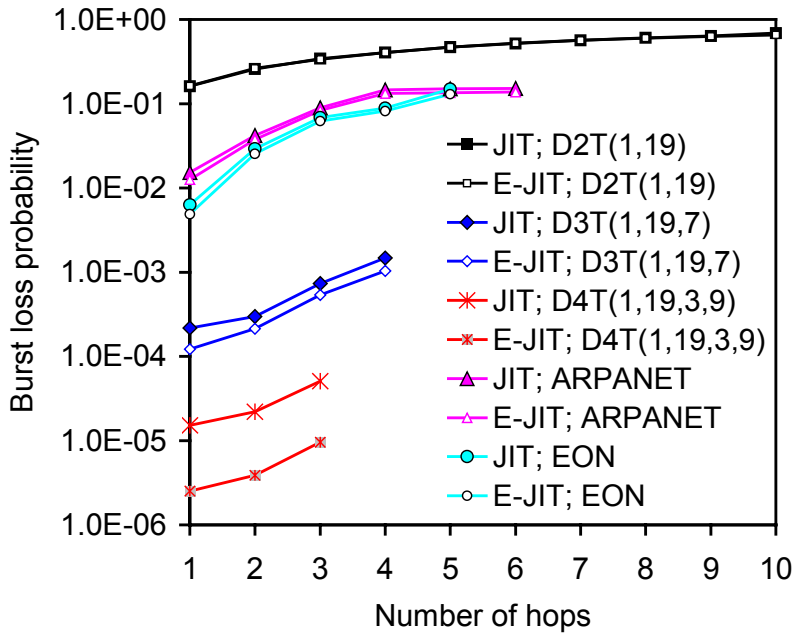


Fig. 5.50. Burst loss probability, as a function of number of number of hops, for ring (D2T(1,19)), chordal ring (D3T(1,19,7) and D4T(1,19,3,9)), ARPANET (*N*=20), and EON (*N*=19) using JIT and E-JIT protocols; λ/μ=32; *F*=64; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

The next figures show burst loss probability as a function of λ/μ for the same network topologies mentioned above. Figure 5.51 illustrates the burst loss probability, as a function of λ/μ, in the last hop of D2T(1,15), D3T(1,15,5), D4T(1,15,5,13), Mesh-torus (*N*=16) and NSFNET (*N*=16), for JIT and E-JIT, with *F*=64 data channels per link. As may be seen, the burst loss probability increases with the increase of λ/μ. The behavior of both topologies is similar for each resource reservation protocols; however, the performance of E-JIT is better than JIT. When λ/μ increases, the difference between the performance JIT and E-JIT decreases. For values of λ/μ less than 32 for chordal rings and Mesh-torus, less than 25.6 for NSFNET, and less than 19.2 for ring, the result of burst loss probability is zero. These observations are confirmed in Figure 5.52 for network topologies around 20 nodes (D2T(1,19), D3T(1,19,7), D4T(1,19,3,9), ARPANET, and EON). When λ/μ increases the burst loss probability also increases, and the difference between the performance of JIT and E-JIT also decreases. However, E-JIT performs better than JIT. For D4T(1,19,3,9), when the burst loss probability is lower, the relative performance of JIT and E-JIT is more significant. For values of λ/μ less than 32 for chordal rings, less than 19.2 for ARPANET and EON, and less than 12.8 for ring, the result of burst loss probability is zero.

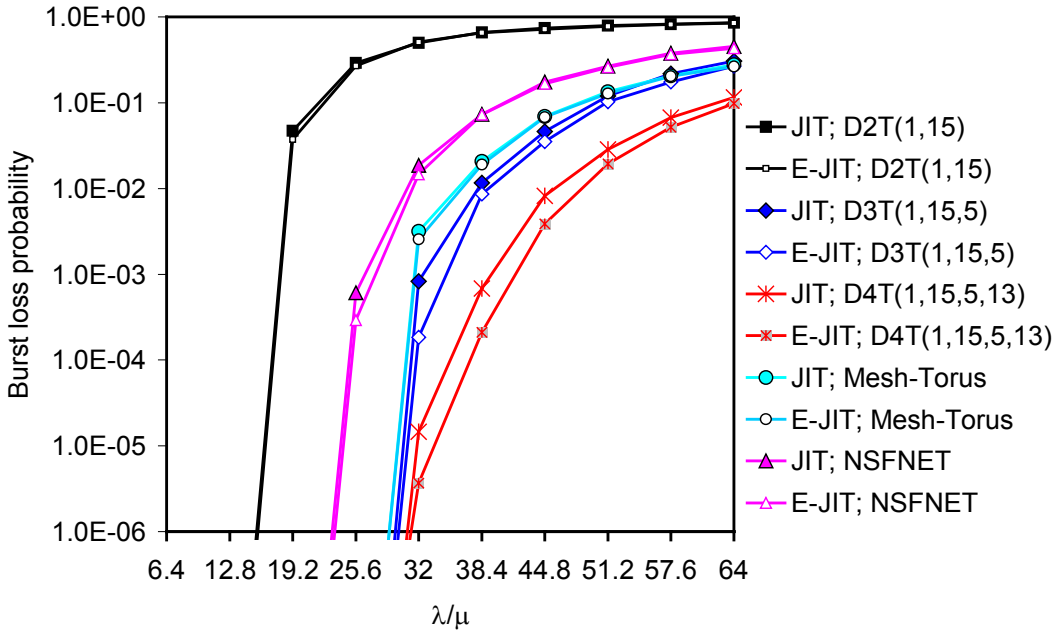


Fig. 5.51. Burst loss probability, as a function of λ/μ, in the last hop ring (D2T(1,15)), chordal ring (D3T(1,15,5) and D4T(1,15,5,13)), NSFNET, and Mesh-Torus networks for JIT and E-JIT protocols; F=64; N=16; λ/μ=32; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

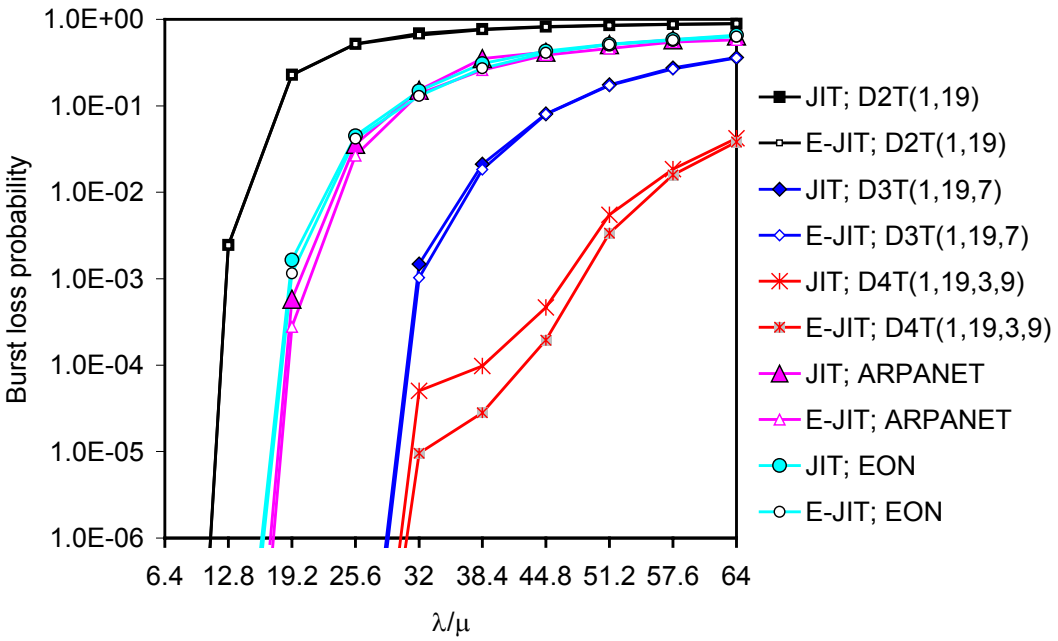


Fig. 5.52. Burst loss probability, as a function of λ/μ, in the last hop of ring (D2T(1,19)), chordal ring (D3T(1,19,7) and D4T(1,19,3,9)), ARPANET (N=20), and EON (N=19) for JIT and E-JIT protocols; F=64; λ/μ=32; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

5.5.2 Impact of Number of Nodes

In this sub-section, the impact of number of nodes in the performance of JIT and E-JIT resource reservation protocols is studied, considering mesh topologies between 10 and 30 nodes. In Figure 5.53, networks with nodal degree around two and three are considered, and Figure 5.44 presents networks with nodal degree around four. Degree-three chordal ring (D3T(1,*N*-1,5)) is used in both figures for comparison purposes. NSFNET with *N*=14 nodes has nodal degree equal to 3, NSFNET with *N*=16 nodes has nodal degree equal to 3.125, and ARPANET with nodal degree equal to 3.2 are examples of nodal degree around three. Mesh-torus with *N*=16 and *N*=25 have nodal degree equal to 4, and European Optical Network (EON) has nodal degree equal to 3.89 are examples of nodal-degree around four. Chordal ring networks with smallest diameter, which present best performance, selected for degree-three network topologies are D3T(1,*N*-1,5) and D3T(1,*N*-1,7), and for degree-four is D4T(1,*N*-1,5,9).

As may be seen in Figure 5.53, for networks with number of nodes less and equal than 18 and equal to 22, D3T(1,*N*-1,5) performs better than others topologies and D3T(1,*N*-1,7) performs better for networks with number of nodes equal to 20 and more than 22 nodes. For networks with nodal degree around three, E-JIT performs better than JIT, but the impact is not significant, except for D3T(1,15,5). Network topologies with nodal degree equal to two (rings and FCCN-NET) present the worst performance. In terms of irregular mesh topologies, NSFNET with 14 nodes performs better than NSFNET with 16 nodes and ARPANET.

In Figure 5.54, degree-four topologies perform better for networks with number of nodes great than 16 nodes. For networks with nodal degree around four, E-JIT performs better than JIT, mainly for degree-four chordal rings with number of nodes great than 16 nodes. In terms of irregular mesh topologies, the performance of Mesh-torus (with *N*=16 and *N*=25 nodes) is very similar and they perform better than EON.

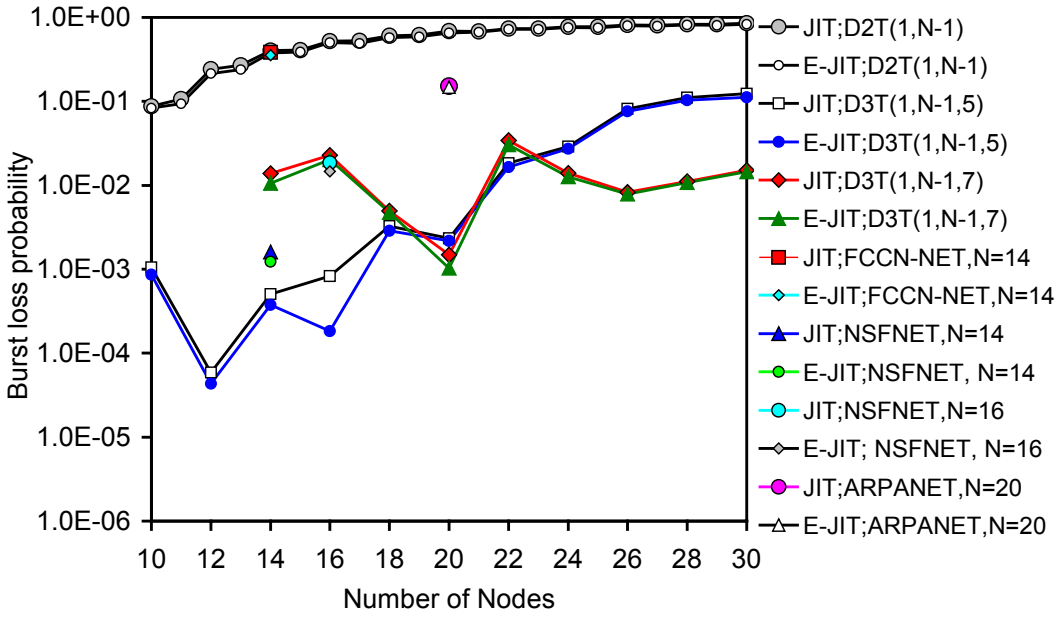


Fig. 5.53. Burst loss probability, as a function of the number of nodes (N), in the last hop of rings, degree-three chordal rings, FCCN-NET, NSFNET, and ARPANET, for JIT and E-JIT protocols; λ/μ=32; F=64; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

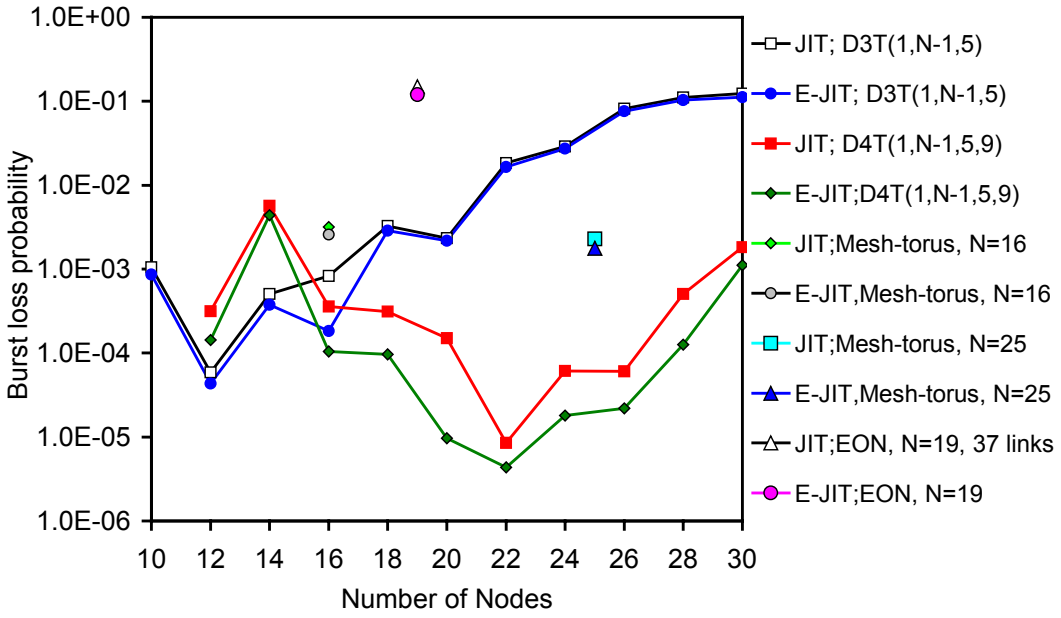


Fig. 5.54. Burst loss probability, as a function of the number of nodes (N), in the last hop of degree-three and degree-four chordal rings, Mesh-torus, and EON, for JIT and E-JIT protocols; λ/μ=32; F=64; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

5.5.3 Influence of Nodal Degree on Nodal Degree Gain

This sub-section focuses on the study of the performance of OBS networks, based on the nodal degree gain, for JIT and E-JIT resource reservation protocols. The performance analysis considers mesh topologies with nodal degree between three and five. Network topologies are grouped in two figures by number of nodes, considering $N \leq 16$ and $N > 20$, respectively, for comparison purposes. D5T(1,15,7,3,9) and D5T(1,19,3,7,11) are the best performance degree-five chordal ring networks with smallest diameter.

Figure 5.55 shows for JIT and E-JIT the nodal degree gain in the last hop of each topology, as a function of the nodal degree, due to the increase of the nodal degree from 2 (D2T(1,14)) to 3 (NSFNET (N=14)), from 2 (D2T(1,15)) to: 3 (D3T(1,15,5)), 3.125 (NSFNET (N=16)), 4 (D4T(1,15,5,13) and Mesh-Torus (N=16)), and 5 (D5T(1,15,7,3,9) with F=64 and λ/μ=32. Figures 5.56 plots the nodal degree gain in the last hop of each topology, as a function of the nodal degree, due to the increase of the nodal degree from 2 D2T(1,18) to 3.89 (EON (N=19)), from 2 (D2T(1,19)) to: 3 (D3T(1,19,7)), 3.2 (ARPANET (N=20)), 4 (D4T(1,19,3,9)), and 5 (D5T(1,19,3,7,11)), and from 2 (D2T(1,24)) to 4 (Mesh-Torus (N=25)) for JIT and E-JIT with F=64 and λ/μ=32. As may be seen in both figures, when the nodal degree increases from 2 to around 3, the largest gain is observed for degree-three chordal rings with N=16 (between slightly less than three and four orders of magnitude) and the smallest gain is observed for the ARPANET (less than one order of magnitude). When the nodal degree increases from 2 to around 4, the largest gain is observed for degree-four chordal rings with N=16 (with a gain between four and five orders of magnitude) and the smallest gain is observed for the EON (with a gain less than one order of magnitude). When the nodal degree increases from 2 to 5, the gain is slightly less than five orders of magnitude for both chordal rings with N=16 and N=20. The largest gain is observed when the nodal degree increases from 2 to 5. In terms of performance comparison of JIT and E-JIT resource reservation protocols, the nodal degree gain of E-JIT is greater than JIT, and the largest gain is observed for degree-three chordal rings with N=16. These results clearly show the importance of the way links are connected in OBS networks, since, in this kind of networks, burst loss probability is a key issue. Examples of this finding are the similar performance of the NSFNET (with N=14 and nodal degree of 3) in comparison with mesh-torus (with N=16 and nodal degree of 4), and the similar performance of the ARPANET (with nodal degree of 3.2) in comparison with EON (with nodal degree of 3.89). These

observations show that more connections between nodes (larger nodal degree) do not mean better performance of those networks.

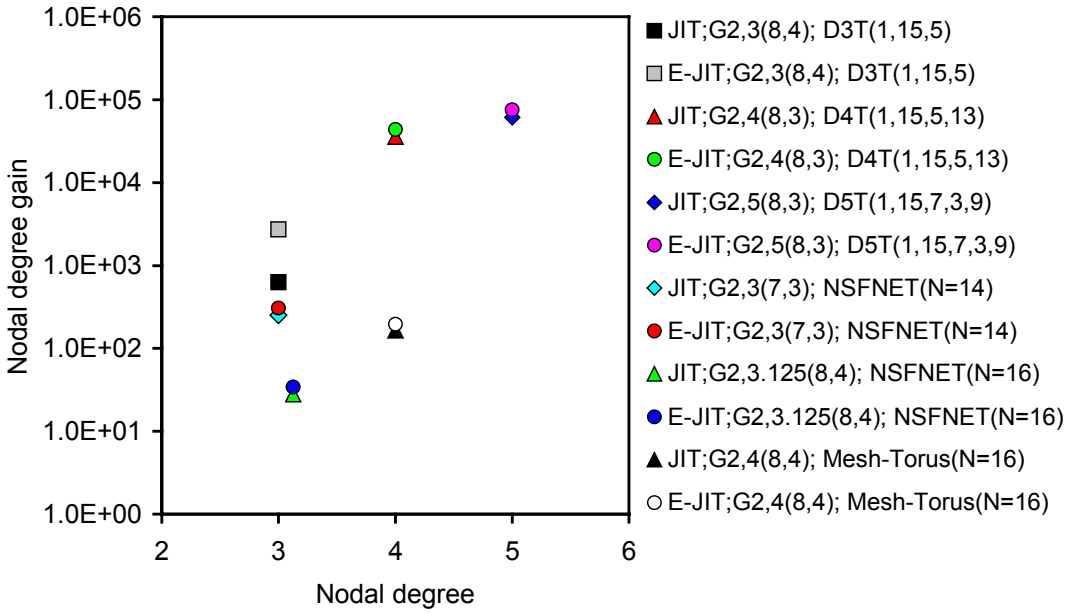


Fig. 5.55. Nodal degree gain in the last hop of each topology (with *N* up to 16), as a function of the nodal degree for JIT and E-JIT protocols; λ/μ=32; *F*=64; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

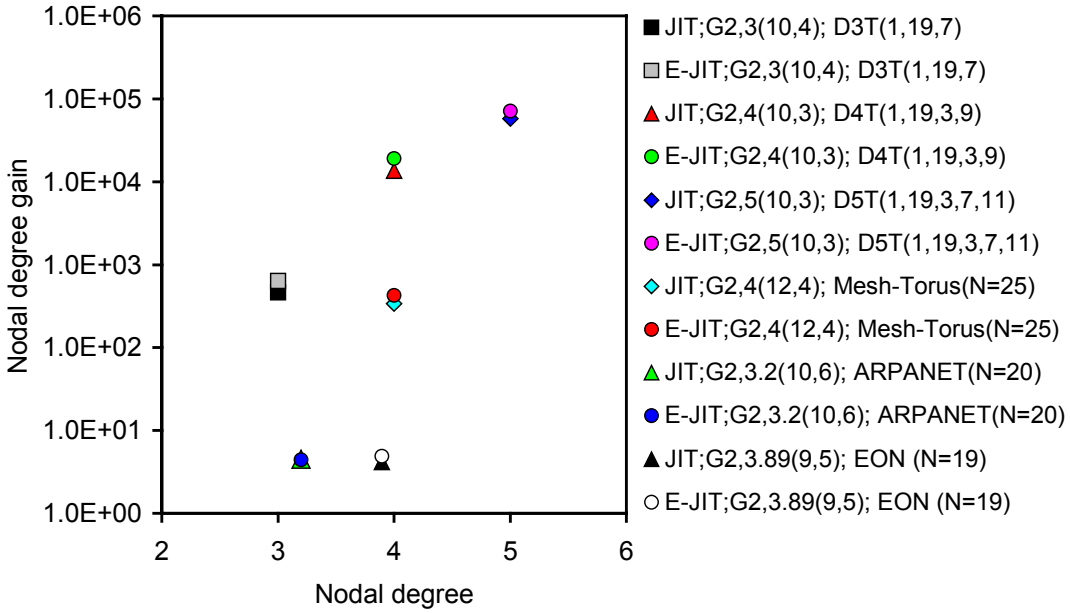


Fig. 5.56. Nodal degree gain in the last hop of each topology (with *N* larger than 16), as a function of the nodal degree for JIT and E-JIT protocols; λ/μ=32; *F*=64; T_{Setup}(JIT)=T_{Setup}(JumpStart)=T_{Setup}(JIT$^+$)=12.5μs; T_{Setup}(JET)=50μs; T_{Setup}(Horizon)=25μs.

5.5.4 Effects of Setup Message Processing Time and OXC Configuration Time

This section studies the influence of setup message processing time (T_{Setup}) and OXC configuration time (T_{OXC}) on the performance of OBS mesh networks for JIT and E-JIT protocols. Network topologies were grouped by nodal degree considering the network topologies with nodal degree around three and nodal degree around four.

Figure 5.57 shows the burst loss probability as a function of OXC configuration time in the last hop of NSFNET (with N=14 and N=16), ARPANET, and D3T(1,19,7) for F=64 and λ/μ=32. In this figure, a fixed value of T_{Setup} time is defined for JIT and E-JIT taking in account the current available technology (JITPAC controllers [141]). T_{OXC} is assumed to range from the value estimated for a near future scenario (T_{OXC}=20μs) up to ten times the value defined for currently available technology, i.e. T_{OXC}=10*10ms=100ms. As may be seen, degree-three chordal ring topology has better performance than other networks, mainly, for $T_{OXC}\leq$10ms. It may also be observed that for $T_{OXC}\leq$1ms, the performance of the two protocols is more or less constant. However it is possible to conclude that despite the improvement and development of new technologies, the network performance does not present enhancement, and T_{OXC} (for values less than 1ms) does not influence the performance of those networks. E-JIT performs slightly better than JIT in every network topologies.

Figure 5.58 illustrates burst loss probability as a function of OXC configuration time in the last hop of NSFNET (with N=14 and N=16), ARPANET, and D3T(1,19,7) for JIT and E-JIT protocols, with F=64 and λ/μ=32. T_{Setup} is assumed to change with T_{OXC}, according to (4.3) for both protocols.

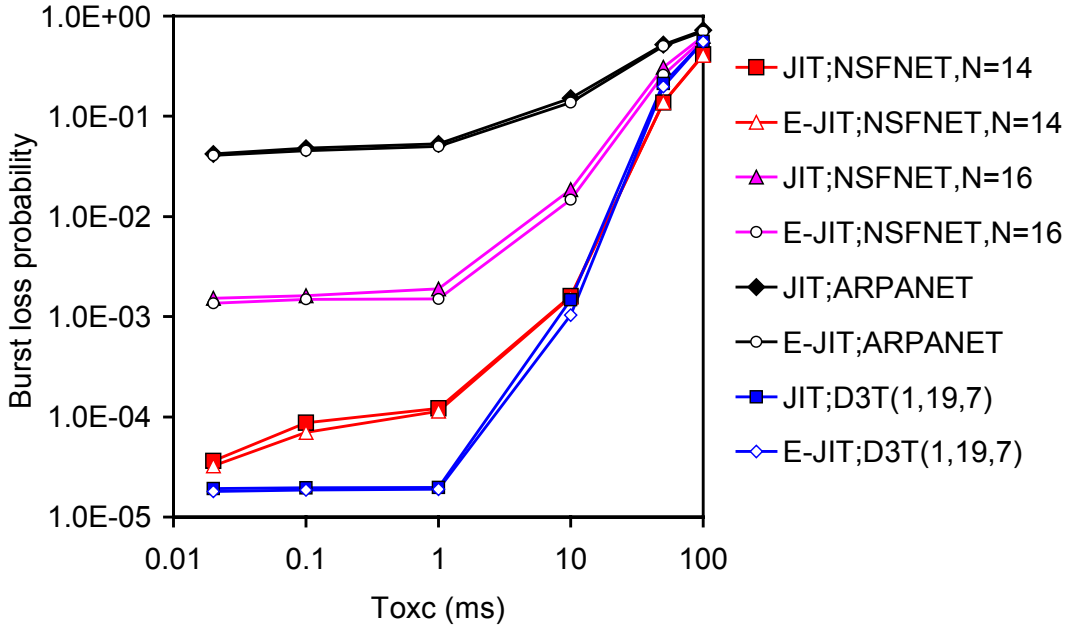


Fig. 5.57. Burst loss probability as a function of OXC configuration time in the last hop of NSFNET (N=14), NSFNET (N=16), ARPANET (N=20), and D3T(1,19,7) (N=20) for JIT and E-JIT; F=64; λ/μ=32; T_{Setup}(JIT)=T_{Setup}(E-JIT)=12.5μs.

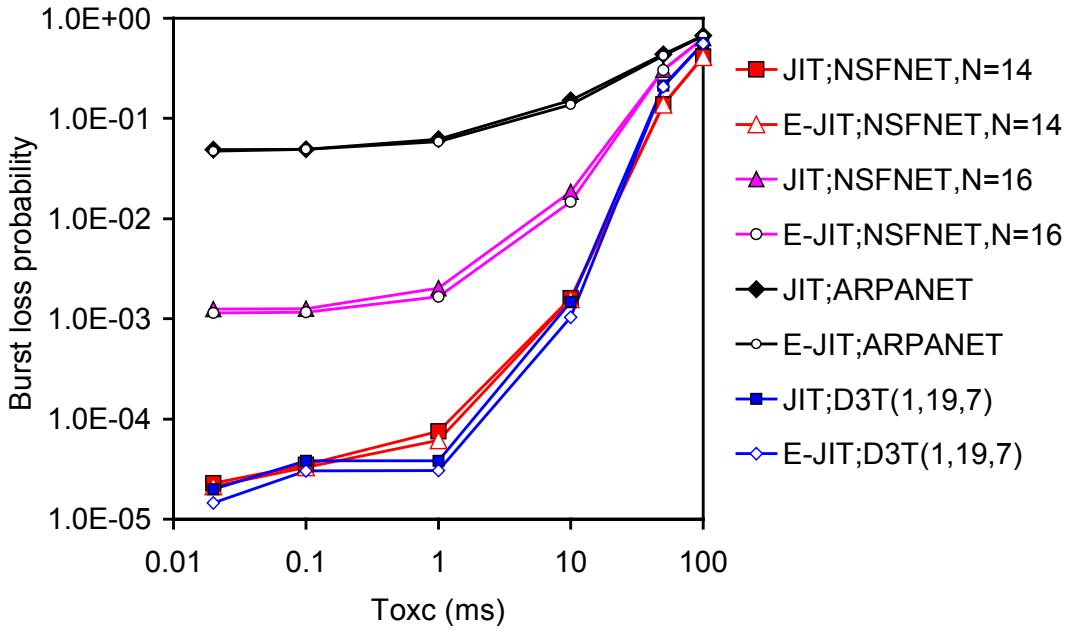


Fig. 5.58. Burst loss probability as a function of OXC configuration time in the last hop of NSFNET (N=14), NSFNET (N=16), ARPANET (N=20), and D3T(1,19,7) (N=20) for JIT and E-JIT; F=64; λ/μ=32; with changing T_{Setup} according to (4.3) for each protocol.

Figure 5.59 plots burst loss probability as a function of setup message processing time in the last hop of the same network topologies shown above in Figures 5.57 and 5.58 for JIT and E-JIT protocols, with F=64 and λ/μ=32. T_{Setup} assumes the value ranging between 1μs and 12.5μs, considering 5μs and 10μs as intermediate times. T_{OXC} is assumed to change with T_{Setup}, being computed according to (4.6).

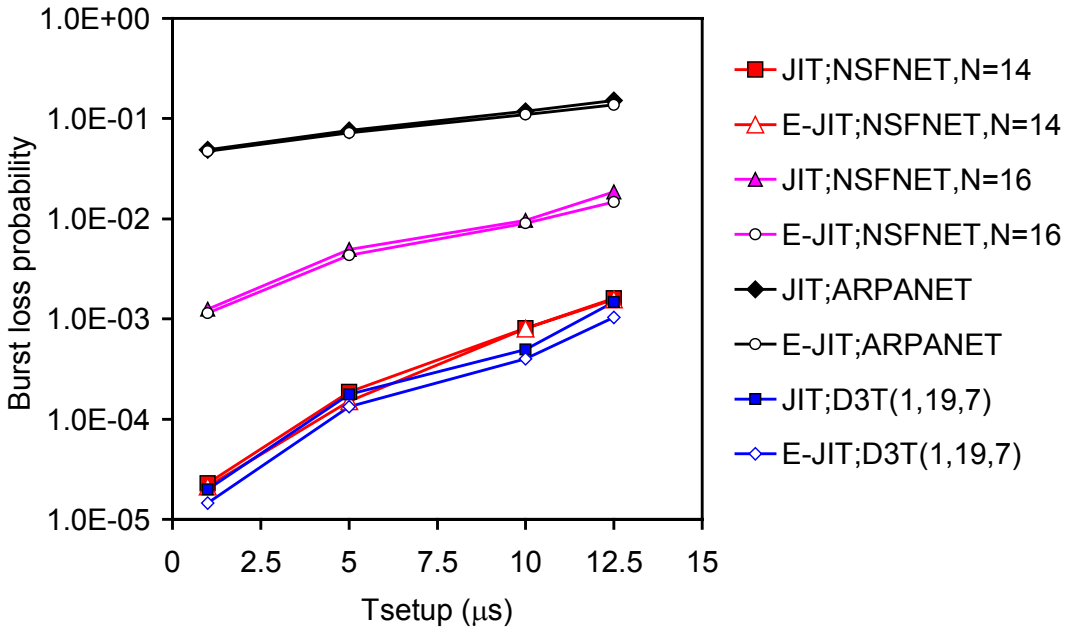


Fig. 5.59. Burst loss probability *versus* T_{Setup} in the last hop of NSFNET (N=14), NSFNET (N=16), ARPANET (N=20), and D3T(1,19,7) (N=20) for JIT and E-JIT; F=64; λ/μ=32, with changing T_{OXC} according to (4.6) for each protocol.

The next figures (Figures 5.60, 5.61, and 5.62) illustrate the same network conditions as previous Figures 5.57, 5.58, and 5.59 for network topologies with nodal degree around four, changing only the value of λ/μ to 44.8. Network topologies under study are mesh-torus (with N=16 and N=25), EON, and the degree-four chordal ring D4T(1,19,3,9). As may be seen in Figures 5.60 and 6.61 the degree-four chordal ring has the best performance and the worst is presented by EON. Mesh-torus presents a similar performance for both number of nodes considered. This result is explained by the same nodal degree (four) and the same way of connections between their nodes. For values of T_{OXC} less than 1ms, the performance of each network does not improve. This observation confirms previous results where the performance of the networks is independent of the change of the T_{OXC}. Additionally, it may also be observed that the relative performance of the both resource reservation protocols is

similar, being E-JIT better than JIT. This result is more evident for degree-four chordal ring when $T_{OXC} \leq 10$ms. These results are confirmed in Figure 5.62.

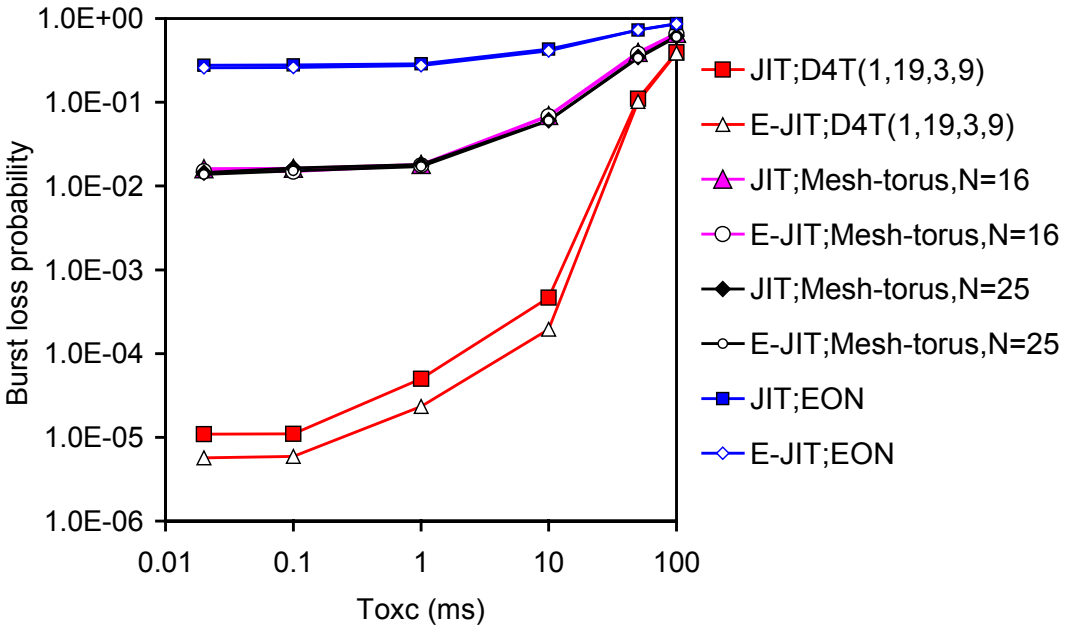


Fig. 5.60. Burst loss probability as a function of OXC configuration time in the last hop of D4T(1,19,3,9) (N=20), Mesh-torus (N=16), Mesh-Torus (N=25), and EON (N=20) for JIT and E-JIT; F=64; λ/μ=44.8; T_{Setup}(JIT)=T_{Setup}(E-JIT)= 12.5μs.

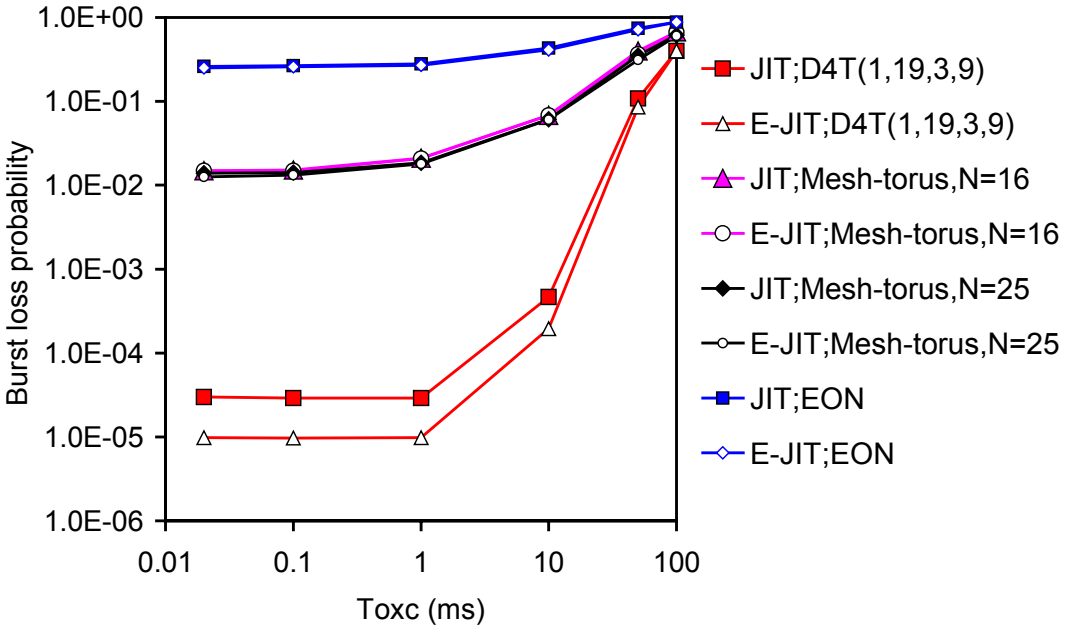


Fig. 5.61. Burst loss probability as a function of OXC configuration time in the last hop of D4T(1,19,3,9) (N=20), Mesh-torus (N=16), Mesh-Torus (N=25), and EON (N=19) for JIT and E-JIT; F=64; λ/μ=44.8; with changing T_{Setup} according to (4.3) for each protocol.

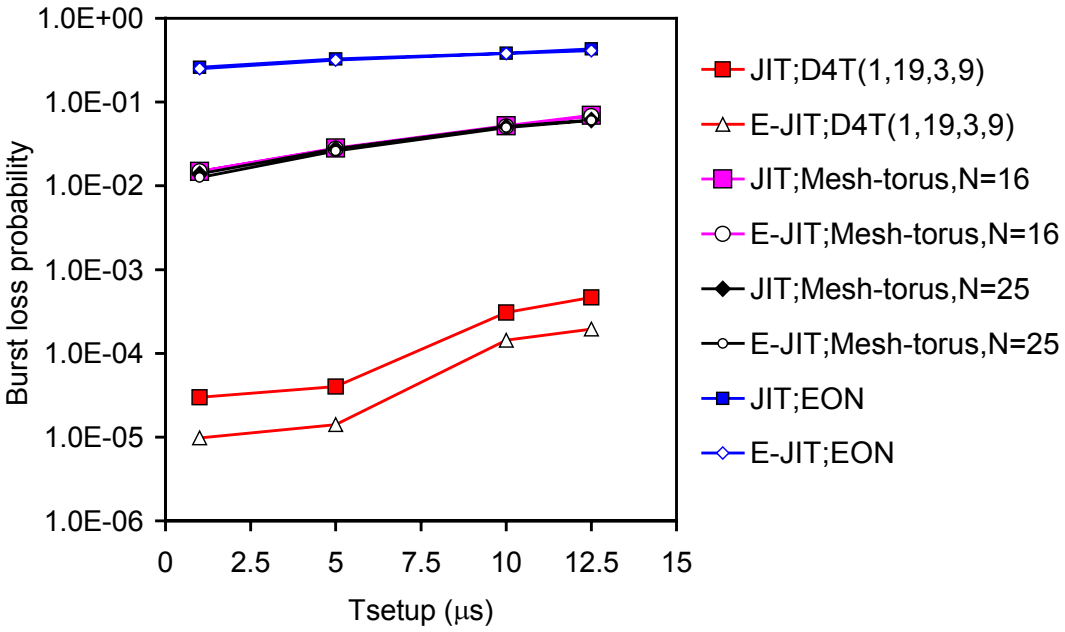


Fig. 5.62. Burst loss probability *versus* T_{Setup} in the last hop of D4T(1,19,3,9) (N=20), Mesh-torus (N=16), Mesh-Torus (N=25), and EON (N=19) for JIT and E-JIT; F=64; λ/μ=44.8, with changing T_{OXC} according to (4.6) for each protocol.

5.6 Conclusions

This chapter analyzed the performance assessment of OBS networks with the following topologies: rings, degree-three, degree-four, degree-five, and degree-six chordal rings, mesh-torus, NSFNET, ARPANET, EON, and the FCCN-NET. The study considers five one-way resource reservation protocols proposed in the research literature: JIT, JumpStart, JIT$^+$, JET, and Horizon, and the new one-way resource reservation protocol proposed in this thesis, called Enhanced Just-in-Time (E-JIT).

First, for network topologies with 14 nodes – D2T(1,13), D3T(1,13,5), NSFNET, and FCCN-NET –, it was observed that FCCN-NET and ring networks have a similar performance in terms of burst loss probability. It was observed that chordal rings with smaller diameter lead to the best network performances. In all of these cases, it was found that the network performance is very close for the five resource reservation protocols under study. For future work, one can conclude for FCCN-NET that, taking into account the geographical extension of Portugal, it is possible to propose another topology based on Refer Telecom network again, but reducing the number of nodes and probably include more links to find alternative routes, including for backup, augmenting the nodal degree of this network topology.

For networks with 16 nodes, the burst loss probability of D2T(1,15), D3T(1,15,5), mesh-torus and NSFNET was analyzed for the existing one-way resource reservation protocols considered. It was shown that chordal rings with smaller diameter lead to the best network performances, i.e., D3T(1,15,5) has the best performance followed by D3T(1,15,7) topology. The performance of the NSFNET is very close to the performance of chordal rings with chord length of w_3=3 or w_3=7. These results reveal the importance of the way links are connected in the network, since chordal rings and NSFNET have similar nodal degrees and therefore a similar number of network links. Also interesting is the fact that chordal rings with w_3=5 (D3T(1,15,5)) have better performance than mesh-torus networks, which have a nodal degree of 4, i.e., 25% more of network links.

Concerning the performance evaluation for number of nodes (N) ranging from N=10 up to N=30 (rings, degree-three and degree-four chordal rings, FCCN-NET, NSFNET, mesh-torus, ARPANET, and EON networks), all results confirm the similar performance of resource reservation protocols with one-way reservation protocols. When the number of nodes is smaller, the D3T(1,N-1,5) has the best performance and when the number of nodes is larger than 16, D4T(1,N-1,5,9) has the best performance.

Section 5.3 discussed performance implications of the nodal degree in OBS mesh networks considered in this chapter. For networks with 16 nodes, it was shown that when the nodal degree increases from 2 to around 3, a larger gain of about three orders of magnitude in the last hop of each topology is observed for chordal ring (D3T(1,15,5)) and a smaller gain between one and two orders of magnitude is observed for the NSFNET. When the nodal degree increases from 2 to around 4, a larger gain between four and five orders of magnitude is observed for degree-four chordal ring (D4T(1,15,5,13)) and a smaller gain between two and three orders of magnitude is observed for the Mesh-torus. When the nodal degree increases from 2 (rings) to 5 (D5T(1,15,7,3,9)) and 6 (D6T(1,15,3,5,7,11)) it is about five orders of magnitude in the last hop of each topology. However, the best performance in terms of nodal degree gain among networks with 16 nodes is the degree-six chordal ring network topology.

The influence of nodal degree on the performance of OBS mesh networks was analyzed for networks with 20 nodes with the following topologies: rings, chordal rings, mesh-torus, NSFNET, ARPANET, and the EON (with N=19 nodes). It was shown that when the nodal degree increases from 2 (ring) to around 3 (D3T(1,19,7)), the largest gain occurs for degree-three chordal rings, being slightly less than three orders of magnitude and the smallest gain occurs for the EON, being the gain less than one order of magnitude. The ARPANET's nodal degree gain is very close to the EON's gain. When

the nodal degree increases from 2 to 4 (D4T(1,19,3,9)), the gain is between one and two orders of magnitude. When the nodal degree increases from 2 to 5 (D5T(1,19,3,7,11)) and 6 (D6T(1,19,3,5,11,15)), the gain is between four and five orders of magnitude. The performance of JIT, JumpSart, JIT$^+$, JET, and Horizon resource reservation protocols is very close.

Concerning the influence of nodal degree on the nodal degree gain, mesh topologies with nodal degree between three and six, and between 14 and 30 nodes are considered: rings, degree-three, degree-four, degree-five, and degree-six chordal rings, mesh-torus (with 16 and 25 nodes), NSFNET (with 14 and 16 nodes), ARPANET and the EON. When the nodal degree increases from 2 to around 3, the largest gain is observed for degree-three chordal rings (slightly less than three orders of magnitude) and the smallest gain is observed for the ARPANET (less than one order of magnitude). When the nodal degree increases from 2 to around 4, the largest gain is observed for degree-four chordal rings (with a gain between four and five orders of magnitude) and the smallest gain is observed for the EON (with a gain less than one order of magnitude). When the nodal degree increases from 2 to around 5 or 6, the gain is between four and six orders of magnitude depending on the number of nodes. The results obtained clearly show the importance of the way links are connected in OBS networks, since large performance differences were observed for the same nodal degree. It was also observed that the performance of the five resource reservation protocols is very close for those topologies, confirming previous results. Since the performance of these protocols is similar, protocols with immediate reservation (JIT, JIT$^+$, and JumpStart) are more suitable for OBS mesh networks, since their implementation is simpler than protocols with delayed reservation.

The effect of setup message processing time (T_{Setup}) and OXC configuration time (T_{OXC}) on the performance of OBS mesh networks was also analyzed. Network topologies were grouped by nodal degree considering the network topologies with nodal degree around three and nodal degree around four. It was observed that for $T_{OXC}\leq$1ms, the performance of mesh topologies is independent of the change of the T_{OXC}, which means that a reduction of the values of T_{OXC} to ones smaller than 1ms does not improve the network performance. For network topologies with nodal degree around three, it was observed that chordal rings (D3T(1,19,7)) and NSFNET with N=14 nodes clearly have better performance than ARPANET and NSFNET with N=16 nodes for $T_{OXC}\leq$50ms. For network topologies with nodal degree around four, the degree-four chordal ring (D4T(1,19,3,9)) has the best performance and the worst is presented by

EON. Mesh-torus presents a similar performance for 16 and 25 nodes. These results are confirmed when T_{Setup} is assumed to change with T_{OXC}, according to (4.3), (4.4), and (4.5). On the other hand, T_{OXC} was assumed to change with T_{Setup}, being computed according to (4.6), (4.7), and (4.8), and it was shown that when the value of T_{Setup} increases, the correspondent burst loss probability also increases. The relative performance of the five existing one-way resource reservation protocols is similar in all cases studied.

The last section was devoted to the performance of E-JIT, in comparison with JIT, taking into account mesh topologies with 16 and 20 nodes. The impact of number of nodes, the influence of nodal degree on nodal degree gain, and the effect of setup message processing time and OXC configuration time were also considered. The performance of E-JIT was compared with JIT to improve the readability of figures since the relative performance of the five existing resource reservation protocols (JIT, JumpStart, JIT$^+$, JET, and Horizon) is very close, as above-mentioned. Furthermore, E-JIT is a JIT based protocol, maintaining its simplicity in terms of implementation. It was shown that E-JIT performs better than JIT. For network topologies with 16 and 20 nodes, when the burst loss probability is lower, the relative better performance of E-JIT over JIT is more significant. Concerning the influence of nodal degree on the nodal degree gain, it was observed that when the nodal degree increases from 2 to around 3, the largest gain is observed for degree-three chordal rings with N=16 (between slightly less than three and four orders of magnitude) and the smallest gain is observed for the ARPANET (less than one order of magnitude). When the nodal degree increases from 2 to around 4, the largest gain is observed for degree-four chordal rings with N=16 (with a gain between four and five orders of magnitude) and the smallest gain is observed for the EON (with a gain less than one order of magnitude). When the nodal degree increases from 2 to 5, the gain is slightly less than five orders of magnitude for both chordal rings with N=16 and N=20. The largest gain is observed when the nodal degree increases from 2 to 5. In terms of performance comparison of JIT and E-JIT protocols, the nodal degree gain of E-JIT is larger than JIT, and the largest gain is observed for degree-three chordal rings with N=16. The effect of setup message processing time and OXC configuration time in the performance of OBS networks for JIT and E-JIT is similar to the other protocols above-mentioned, being the performance of E-JIT slightly better than JIT, mainly when burst loss probability is smaller.

Chapter 6

Conclusions

Throughout this thesis the performance of one-way resource reservation protocols in IP over optical burst switched networks was studied. This chapter presents a synthesis of the main achievements and points to several directions for future work.

After introducing and delimiting the theme of the thesis, describing the objectives, and showing its main contributions, in Chapter 2, the state of the art of OBS networks was presented. This chapter began with the description of the OBS network architecture giving special attention to the edge and core nodes. After, it focused on the main tasks performed by edge nodes, mainly, the burst assembly process and its most relevant algorithms. Concerning core nodes, OBS resource reservation protocols were described, including the proposal of a new one-way resource reservation protocol, called Enhanced Just-in-Time (E-JIT), and the problem of contention resolution addressing several proposals to solve it. Other OBS issues were also considered, namely, the quality of service, transmission control protocol (TCP) over OBS, multicasting, burst grooming, and OBS applications.

Chapter 3 presented objectives, design, implementation and validation of a simulator for OBS networks, named OBSim. After having studied the existing methodologies for performance evaluation of OBS networks (mainly, analytical models and simulators), the need of developing a new simulator from scratch was identified. The developed simulator implemented a model of OBS networks based on objects, which allows the estimation of the performance of a given OBS network

defined by the user. The results of the simulator have been validated, and thus the simulator may be used as a tool to estimate the performance of the OBS networks.

All the mesh network topologies studied in this thesis and used to evaluate the performance of OBS networks, both regulars and irregulars, were also presented. Considered regular topologies were rings, chordal-rings (with nodal degree three, four, five, and six), and mesh-torus (with N=16 nodes and 32 links, and with N=25 nodes and 50 links). In terms of irregular topologies, the following were considered: NSFNET (with N=14 nodes and N=16 nodes), ARPANET, European Optical Network, and the proposal of the backbone for Portuguese FCCN (FCCN-NET).

In Chapter 4, the performance assessment of JIT, JumpStart, JIT$^+$, JET, and Horizon one-way resource reservation protocols in OBS networks with ring and chordal ring topologies were analyzed. In terms of chordal rings, it was focused on degree-three and degree-four chordal rings given the proximity with nodal degree of real network topologies such as NSFNET, ARPANET or European Optical Network. First, the burst loss probability, the nodal degree gain, and the chord length gain were defined. Burst loss probability is a very important performance metric to evaluate the performance of OBS networks and it was defined as the probability that a burst transmission does not arrive at its destination. Nodal degree gain was proposed in order to quantify the benefits due to the increase of nodal degree while the chord length gain allows to quantify the benefits due to a better choice of the chord length.

The study was focused on the performance of degree-three chordal ring networks with 20 nodes, and it was shown that for a network with 20 nodes the best performance was obtained for D3T(1,19,7). It was revealed that the nodal degree gain due to the increase of nodal degree from two (D2T(1,19)) to three (D3T(1,19,7)) is about three orders of magnitude in the first hop of both topologies, and is between two and three orders of magnitude in the last hop of each topology. It was also shown that the largest chord length gain is slightly less than two orders of magnitude, due to the choice of the chord length of w_3=7 instead of a chord length of w_3=5. Concerning the study of the effect of the setup message processing time (T_{Setup}) and the OXC configuration time (T_{OXC}), it was observed that for $T_{OXC}\leq$1ms, the performance of the chordal ring is independent of the change of the T_{OXC}, which means that a reduction of the values of T_{OXC} to ones smaller than 1ms does not improve the network performance. It was also observed that for degree-three chordal rings, a reduction of the T_{OXC} from 10ms down to 20μs leads to a performance improvement of about two orders of magnitude. For rings, the burst loss

is so high that the reduction of T_{OXC} does not have impact on the network performance. After, a performance analysis of OBS degree-three and degree-four chordal ring networks for the five existing resource reservation protocols under study was presented. For comparison purposes, ring topologies were also considered. It was shown that, for a network with 20 nodes, degree-four chordal ring topologies with smallest diameter have better performance regarding other topologies with the same nodal degree and different chord length (w_4). The nodal degree gain due to the increase of nodal degree from two (ring) to three (degree-three chordal ring) was between two and three orders of magnitude in the last hop of each topology. It was also shown that the nodal degree gain due to the increase of nodal degree from two to four (degree-four chordal ring) is around four orders of magnitude. The effect of setup message processing time (T_{Setup}) and OXC configuration time (T_{OXC}) was presented. The reduction of T_{Setup} performs slightly better for degree-four topology, mainly, for T_{OXC}=20μs, and with the reduction of T_{OXC} the network performs better. It was observed that the influence of the nodal degree in the performance of a given chordal ring topology is larger than the influence of T_{Setup} and T_{OXC}. This study confirms previous observations concluding that, for 20 nodes, degree-four chordal ring performs better than degree-three and the worst performance is presented by ring. In all the studied cases it was observed that the performances of the JIT, JumpStart, JIT+, JET and Horizon resource reservation protocols are very close.

In Chapter 5, first, a performance assessment of JIT, JumpStart, JIT$^+$, JET, and Horizon resource reservation protocols in OBS networks with mesh topologies were discussed and, later, a performance evaluation of E-JIT protocol was presented. The analysis was focused on OBS networks with the following mesh topologies: chordal rings with nodal-degree between three and six and number of nodes ranging from 10 up to 30, mesh-torus with 16 and 25 nodes, the NSFNET with 14-nodes and 21 links, the NSFNET with 16 nodes and 25 links, the ARPANET with 20 nodes and 32 links, the European Optical Network (EON) with 19 nodes and 37 links, and the Portuguese *Fundação para a Computação Científica Nacional* network (FCCN-NET) with 14 nodes and 14 links.

For network topologies with 14 nodes, it was observed that FCCN-NET and ring networks have a similar performance in terms of burst loss probability, and chordal rings with smallest diameter lead to best network performances. For networks with 16 nodes it was shown that chordal rings with smallest diameter lead to best network performances, i.e., D3T(1,15,5) has the best performance followed by D3T(1,15,7) topology. The performance of the NSFNET is very close to the performance of chordal

rings with chord length of w_3=3 or w_3=7. These results reveals the importance of the way links are connected in the network, since chordal rings and NSFNET have similar nodal degrees and therefore a similar number of network links. Also interesting is the fact that chordal rings with w_3=5 (D3T(1,15,5)) have better performance than mesh-torus networks, which have a nodal degree of 4, i.e., 25% more of network links.

Concerning the performance evaluation for the networks under study in this thesis for number of nodes (*N*) ranging from 10 up to 30, all results confirm the similar performance of the five existing one-way resource reservation protocols studied in this thesis. When the number of nodes is smaller, the D3T(1,*N*-1,5) has the best performance and when the number of nodes is larger than 16, D4T(1,*N*-1,5,9) has the best performance.

The performance implications of the nodal degree in OBS mesh networks were also studied. For networks with 16 nodes, it was shown that when the nodal degree increases from 2 to around 3, a larger gain of about three orders of magnitude in the last hop of each topology is observed for chordal ring (D3T(1,15,5)) and a smaller gain between one and two orders of magnitude is observed for the NSFNET. When the nodal degree increases from 2 to around 4, a larger gain between four and five orders of magnitude is observed for degree-four chordal ring (D4T(1,15,5,13)) and a smaller gain between two and three orders of magnitude is observed for the Mesh-torus. When the nodal degree increases from 2 (rings) to 5 (D5T(1,15,7,3,9)) and 6 (D6T(1,15,3,5,7,11)) it is about five orders of magnitude in the last hop of each topology. However, the best performance in terms of nodal degree gain for all networks with 16 nodes considered is the degree-six chordal ring network topology.

The impact of nodal degree on the performance of OBS mesh networks was analyzed for networks with 20 nodes and it was observed that when the nodal degree increases from 2 (ring) to around 3 (D3T(1,19,7)), the largest gain occurs for degree three chordal rings, being slightly less than three orders of magnitude and the smallest gain occurs for the EON, being the gain less than one order of magnitude. The nodal degree gain of the ARPANET is very close to the EON gain. When the nodal degree increases from 2 to 4 (D4T(1,19,3,9)), the gain is between one and two orders of magnitude. When the nodal degree increases from 2 to 5 (D5T(1,19,3,7,11)) and 6 (D6T(1,19,3,5,11,15)), the gain is between four and five orders of magnitude.

To study the influence of nodal degree on the nodal degree gain, mesh topologies with nodal degree between three and six and with number of nodes ranging from 14 up to 30 were considered. When the nodal degree increases from 2 to around 3, the largest gain is observed for degree-three chordal rings (slightly less

than three orders of magnitude) and the smallest gain is observed for the ARPANET (less than one order of magnitude). When the nodal degree increases from 2 to around 4, the largest gain is observed for degree-four chordal rings (with a gain between four and five orders of magnitude) and the smallest gain is observed for the EON (with a gain less than one order of magnitude). When the nodal degree increases from 2 to around 5 or 6, the gain is between four and six orders of magnitude depending on the number of nodes. The results obtained clearly show the importance of the way links are connected in OBS networks, since large performance differences were observed for the same nodal degree. It was also observed that the performance of the five resource reservation protocols is very close for those topologies, confirming previous results. Since the performance of these protocols is similar, protocols with immediate reservation (JIT, JIT$^+$, and JumpStart) are more suitable for OBS mesh networks, since their implementation is simpler than protocols with delayed reservation.

The influence of setup message processing time (T_{Setup}) and OXC configuration time (T_{OXC}) on the performance of OBS mesh networks was also analyzed. Network topologies were grouped by nodal degree considering the network topologies with nodal degree around three and nodal degree around four. It was observed that for $T_{OXC}\leq$1ms, the performance of mesh topologies is independent of the change of the T_{OXC}, which means that a reduction of the values of T_{OXC} to ones smaller than 1ms does not improve the network performance. For network topologies with nodal degree around three, it was observed that chordal rings (D3T(1,19,7)) and NSFNET with N=14 nodes clearly have better performance than ARPANET and NSFNET with N=16 nodes for $T_{OXC}\leq$50ms. For network topologies with nodal degree around four, the degree-four chordal ring (D4T(1,19,3,9)) has the best performance and the worst is observed for EON. Mesh-torus presents a similar performance for 16 and 25 nodes. These results are confirmed when T_{Setup} is assumed to change with T_{OXC}, according to equations (4.3), (4.4), and (4.5). On the other hand, T_{OXC} was assumed to change with T_{Setup}, being computed according to (4.6), (4.7), and (4.8), and it was shown that when the value of T_{Setup} increases, the correspondent burst loss probability also increase.

In the last section of Chapter 5 the performance of E-JIT was studied, in comparison with JIT, taking into account mesh topologies with 16 and 20 nodes. The impact of number of nodes, the influence of nodal degree on nodal degree gain, and the effect of setup message processing time and OXC configuration time were also considered. The performance of E-JIT was compared with JIT to improve the

readability of figures, as the relative performance of the five existing resource reservation protocols (JIT, JumpStart, JIT$^+$, JET, and Horizon) is very close, as above-mentioned. Furthermore, E-JIT is a JIT based protocol, maintaining its simplicity in terms of implementation. It was shown that E-JIT performs better than JIT. For network topologies with 16 and 20 nodes, it was observed that when the burst loss probability is smaller, the relative performance of E-JIT is more significant. Concerning the influence of nodal degree on the nodal degree gain, it was observed that when the nodal degree increases from 2 to around 3, the largest gain is observed for degree-three chordal rings with N=16 (between slightly less than three and four orders of magnitude) and the smallest gain is observed for the ARPANET (less than one order of magnitude). When the nodal degree increases from 2 to around 4, the largest gain is observed for degree-four chordal rings with N=16 (with a gain between four and five orders of magnitude) and the smallest gain is observed for the EON (with a gain less than one order of magnitude). When the nodal degree increases from 2 to 5, the gain is slightly less than five orders of magnitude for both chordal rings with N=16 and N=20. The largest gain is observed when the nodal degree increases from 2 to 5. In terms of performance comparison of JIT and E-JIT protocols, the nodal degree gain of E-JIT is greater than JIT, and the largest gain is observed for degree-three chordal rings with N=16. The effect of setup message processing time and OXC configuration time is similar to the other above-mentioned protocols, being the performance of E-JIT slightly better than JIT, mainly when burst loss probability is smaller.

The main objective of this thesis was to present a performance study of OBS networks with ring and mesh topologies for the most important five one-way resource reservation protocols (JIT, JumpStart, JIT$^+$, JET, and Horizon). This was carried out taking into account the burst offset length, the edge to core node delay, the propagation delay on each link between core nodes, the number of data channels per link, the processing time of setup messages, the optical switch configuration time, the network size, and the topology. This objective was successfully accomplished. Furthermore, as a result of this research, it was possible to develop an optimization of the operation of JIT, which led to the proposal of a new one-way resource reservation protocol, termed Enhanced JIT (E-JIT). These findingsare likely to be of relevance in the field of the development of next generation optical Internet networks.

6.1 Future Work

To conclude this thesis, it remains to suggest future research directions that result from this research work:

- To study QoS issues related with the one-way resource reservation protocols studied in this thesis (JIT, JumpStart, JIT+, JET Horizon, and E-JIT). For example, introducing different priorities for different classes of bursts.
- To study the effect of wavelength converters in the performance of the one-way resource reservation protocols studied in this thesis.
- To study different configurations of FCCN-NET, namely, reducing number of core nodes and, mainly, the number of edge nodes per core.
- To implement and to evaluate the performance of E-JIT in a testbed or in a real OBS network.

References

[1] J. S. Turner, "Terabit burst switching", *Journal of High Speed Networks*, Vol. 8, no. 1, January 1999, pp. 3-16.

[2] J. Y. Wei, J. L. Pastor, R. S. Ramamurthy, and Y. Tsai, "Just-in-time optical burst switching for multiwavelength networks", Proceedings of 5th IFIP TC6 International Conference on Broadband Communications (BC '99), Hong Kong, Kluwer Academic Publisher, pp. 339-352, November, 1999.

[3] H. M. Chaskar, S. Verma, and R. Ravikanth, "A framework to support IP over WDM using optical burst switching", Proceedings of IEEE/ACM/SPIE Optical Networks Workshop, January 5, 2000.

[4] K. Dolzer, C. Gauger, J. Späth, and S. Bodamer, "Evaluation of reservation mechanisms for optical burst switching", *AEÜ International Journal of Electronics and Communications*, Vol. 55, no. 1, January 2001

[5] S. Verma, H. Chaskar, and R. Ravikanth, "Optical Burst Switching: A Viable Solution for Terabit IP Backbone", *IEEE Network*, Vol. 14, no. 6, November/December 2000, pp. 48-53.

[6] K. Dolzer and C. M. Gauger, "On Burst Assembly in Optical Burst Switching Networks – A Performance Evaluation of Just-Enough-Time", Proceedings of 17th International Teletraffic Congress, Salvador, Brazil, pp. 149-161, September 24-28, 2001.

[7] S. Sengupta and R. Ramamurthy, "From network design to dynamic provisioning and restoration in optical cross-connect mesh networks: an architectural and algorithmic overview", *IEEE Network*, Vol. 15, no. 4, July-August 2001, pp. 46-54.

[8] P.-H. Ho and H. T. Mouftah, "A framework for service-guaranteed shared protection in WDM mesh networks", *IEEE Communications Magazine*, Vol. 40, no. 2, February 2002, pp. 97-103.

[9] K. Zhu and B. Mukherjee, "Traffic Grooming in an Optical WDM Mesh Network", *IEEE Journal on Selected Areas in Communications*, Vol. 20, no. 1, January 2002, pp. 122-133.

[10] S. Yao, F. Xue, B. Mukherjee, S. J. B. Yoo, and S. Dixit, "Electrical Ingress Buffering and Traffic Aggregation for Optical Packet Switching and Their Effect on TCP-Level Performance in Optical Mesh Networks", *IEEE Communications Letters*, Vol. 40, no. 9, September 2002, pp. 66-72.

[11] M. Oh, "Network Management Agent Allocation Scheme in Mesh Networks", *IEEE Communications Letters*, Vol. 7, no. 12, December 2003, pp. 601-603.

[12] C. Assi, Y. Ye, S. Dixit, and M. Ali, "Control and Management Protocols for Survivable Optical Mesh Networks", *IEEE Journal of Lightwave Technology*, Vol. 21, no. 11, November 2003, pp. 2638-2651.

[13] N. K. Singhal, L. H. Sahasrabuddhe, and B. Mukherjee, "Provisioning of Survivable Multicast Sessions Against Single Link Failures in Optical WDM Mesh Networks", *IEEE Journal of Lightwave Technology*, Vol. 21, no. 11, November 2003, pp. 2587-2594.

[14] S.-i. Kim and S. S. Lumetta, "Restoration of All-Optical Mesh Networks With Path-Based Flooding", *IEEE Journal of Lightwave Technology*, Vol. 21, no. 11, November 2003, pp. 2605-2616.

[15] N. K. Singhal and B. Mukherjee, "Protecting Multicast Sessions in WDM Optical Mesh Networks", *IEEE Journal of Lightwave Technology*, Vol. 21, no. 4, April 2003, pp. 884-892.

[16] C. S. Ou, J. Zhang, H. Zang, L. H. Sahasrabuddhe, and B. Mukherjee, "New and Improved Approaches for Shared-Path Protection in WDM Mesh Networks", *IEEE Journal of Lightwave Technology*, Vol. 22, no. 5, May 2004, pp. 1223-1232.

[17] J. J. P. C. Rodrigues, M. M. Freire, P. P. Monteiro, and P. Lorenz, "Optical Burst Switching: A New Switching Paradigm for High-Speed Internet," in *Encyclopedia of Multimedia Technology and Networking*, M. Pagani (Ed.), Vol. II, Idea Group Reference, ISBN: 1-59140-561-0, 2005.

[18] J. J. P. C. Rodrigues, N. M. Garcia, M. M. Freire, and P. Lorenz, "Object-Oriented Modeling and Simulation of Optical Burst Switching Networks", 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, TX, USA, IEEE Press, ISBN: 0-7803-8798-8, pp. 288-292, November 29 - December 03, 2004.

[19] J. J. P. C. Rodrigues and M. M. Freire, "Object-oriented Modeling and Simulation of IP over Optical Burst Switched Mesh Networks," in *Simulation and Modeling: Current Technologies and Applications*, A. A. E. Sheikh, A. T. A. Ajeeli, and E. M. Abu-Taieh (Eds.), Idea Group, Inc., Accepted for publication, 2006.

[20] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "Performance Assessment of Optical Burst Switching Ring and Chordal Ring Networks", *Telecommunications Systems*, Vol. 27, no. 2-4, October - December 2004, pp. 133-149.

[21] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "Performance Comparison of Optical Burst Switching Ring and Chordal Ring Networks Using Just-in-Time and Just-Enough-Time Signaling Protocols", Proceedings of 3rd International Conference on Networking (ICN'04), Gosier, Guadalupe, French Caribbean, ISBN: 0-86341-325-0, pp. 65-69, March 1-4, 2004.

[22] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "Performance Assessment of Optical Burst Switched Degree-Four Chordal Ring Networks", Proceedings of 11th International Conference on Telecommunications (ICT'04), J. N. d. Souza, P. Dini, and P. Lorenz (Eds.), Lecture Notes in Computer Science (LNCS), Vol. 3124, Fortaleza, Brazil, Springer, ISBN: 3-540-22571-4, pp. 760-765, August 01 - 06, 2004.

[23] J. J. P. C. Rodrigues and M. M. Freire, "Performance Assessment of Signaling Protocols in Optical Burst Switching Mesh Networks", Proceedings of The International Conference on Information Networking (ICOIN 2004) - "Convergence in Broadband and Mobile Networking", Vol. III, Busan, Korea, Korea Information Science Society, pp. 1217-1226, February 18-20, 2004.

[24] J. J. P. C. Rodrigues and M. M. Freire, "Performance Assessment of Signaling Protocols in Optical Burst Switching Mesh Networks," in *Information Networking. Netwosrking Technologies for Broadband and Mobile Networks*, H.-K. Kahng (Ed.), *Lecture Notes in Computer Science (LNCS)*, Vol. 3090, Revised Selected Papers of ICOIN 2004, Springer-Verlag, Berlin Heidelberg, 2004, pp. 750-759.

[25] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "IP Over Optical Fibre Infrastructure: a Performance Assessment of the Portuguese Backbone", Proceedings of 5th Conference on Telecommunications (CONFTELE 2005), D.

M. Santos (Ed.), CD-ROM, Tomar, Portugal, Instituto de Telecomunicações, ISBN: 972-98115-9-8, April 06 - 08, 2005.

[26] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "One-way Resource Reservation Protocols for IP Over Optical Burst Switched Mesh Networks", Proceedings of International Conference on Systems Communications 2005, P. Dini, P. Lorenz, S. Soulhi, S. Cherkaoui, D. Mynbaev, J. J. Rodrigues, A. Hafid, H.-J. Zepernick, and J. Zheng (Eds.), Montreal, Canada, IEEE Computer Society, ISBN: 0-7695-2422-2, pp. 229-234, August, 14-17, 2005.

[27] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "Performance Assessment of Signaling Protocols with One-Way Reservation Schemes for Optical Burst Switching Networks", Proceedings of 7th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC'04), Z. Mammeri and P. Lorenz (Eds.), Lecture Notes in Computer Science (LNCS), Vol. 3079, Toulouse, France, Springer, pp. 821-831, June 30 - July 3, 2004.

[28] J. J. P. C. Rodrigues, N. M. Garcia, and M. M. Freire, "Análise de Desempenho de Protocolos de Sinalização em Redes com Comutação Óptica de Agregados de Pacotes", Proceedings of 7ª Conferência sobre Redes de Computadores (CRC'2004), A. Santos and P. Veiga (Eds.), CD-ROM, Leiria, Portugal, 7 e 8 de Outubro, 2004.

[29] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "Performance Implications of Nodal Degree for Optical Burst Switching Mesh Networks Using Signaling Protocols with One-way Reservation Schemes", Proceedings of The International Conference on Information Networking (ICOIN 2005), C. Kim (Ed.) Lecture Notes in Computer Science (LNCS), LNCS 3391/2005, Jeju, Korea, Springer, ISBN: 3-540-24467-0, pp. 352-361, January 31 - February 2, 2005.

[30] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "The Role of Meshing Degree in Optical Burst Switching Networks Using Signaling Protocols with One-way Reservation Schemes", Proceedings of 4th International Conference on Networking (ICN'2005), P. Lorenz and P. Dini (Eds.), Lecture Notes in Computer Science (LNCS), Vol. 3420, Reunion Island, Springer, ISBN: 3-540-25339-3, pp. 44-51, April 17 - 23, 2005.

[31] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "Performance Implications of Meshing Degree for Optical Burst Switched Networks Using One-way Resource Reservation Protocols", *Telecommunications Systems*, no., accepted for publication, 2005

[32] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "Impact of Setup Message Processing and Optical Switch Configuration Times on the Performance of IP Over Optical Burst Switching Networks", Proceedings of Computer and Information Sciences - ISCIS 2005, P. Yolum (Ed.) Lecture Notes in Computer Science (LNCS), Vol. 3733, Istambul, Turkey, Springer-Verlag, pp. 264-273, October 26 - 28, 2005.

[33] J. J. P. C. Rodrigues, M. M. Freire, N. M. Garcia, and P. P. Monteiro, "Technical Report," March 2006.

[34] R. Ramaswani and K. N. Sivarajan, *Optical Networks*, Academic Press, San Diego, 2002.

[35] B. S. Arnaud and G. Xiao, "The third revolution in optical networking", *Optical Networks Magazine*, Vol. 2, no. 2, 2001, pp. 19-20.

[36] P. Tomsu and C. Schmutzer, *Next generation optical networks*, Prentice Hall, Upper Saddle River, New Jersey, 2002.

[37] K. H. Liu, *IP Over WDM*, John Wiley & Sons, West Sussex, 2002.

[38] S. S. Dixit, *IP over WDM: Building the Next Generation Optical Internet*, John Wiley & Sons, West Sussex, 2003.

[39] I. Chlamtac, A. Ganz, and G. Karm, "Lightpath Communications: An Approach to High Bandwidth Optical WAN's", *IEEE Transactions on Communications*, Vol. 40, no. 7, July 1992, pp. 1171-1182.

[40] B. Mukherjee, S. Ramamurthy, D. Banerjee, and A. Mukherjee, "Some Principles for Designing a Wide-Area Optical Network", Proceedings of IEEE Infocom, IEEE (Ed.), Toronto, Canada, pp. 110-118, 1994.

[41] R. Ramaswami and K. N. Sivarajan, "Optimal Routing and Wavelength Assignment in All-Optical Networks", Proceedings of IEEE Infocom, Toronto, Canada, IEEE, pp. 970-979, 1994.

[42] T. E. Stern and K. Bala, *Multiwavelength Optical Networks - A Layered Approach*, Addison-Wesley, Reading, 1999.

[43] C. Qiao and M. Yoo, "A Taxonomy of Switching Techniques," in *Optical WDM Networks: Principles and Practice*, K. M. Sivalingam and S. Subramaniam (Eds.), Kluwer Academic Publishers, 2000, pp. 103-125.

[44] I. Chlamtac, A. Fumagalli, L. G. Kazovsky, P. Melman, W. H. Nelson, P. Poggiolini, M. Cerisola, A. N. M. M. Choudhury, T. K. Fong, R. T. Hofmeister, C.-L. Lu, A. Mekkittikul, D. J. M. S. IX, C.-J. Suh, and E. W. M. Wong, "CORD:

Contention Resolution by Delay Lines", *IEEE Journal on Selected Areas in Communications*, Vol. 14, no. 5, June 1996, pp. 1014-1029.

[45] R. L. Cruz and J.-T. Tsai, "COD: Alternative Architectures for High Speed Packet Switching", *IEEE / ACM Transactions on Networking*, Vol. 4, no. 1, February 1996, pp. 11-21.

[46] S. Yao, B. Mukherjee, and S. Dixit, "Advances in Photonic Packet Switching: An Overview", *IEEE Communications Magazine*, Vol. 38, no. 2, February 2000, pp. 84-94.

[47] L. Xu, H. G. Perros, and G. N. Rouskas, "Techniques for Optical Packet Switching and Optical Burst Switching", *IEEE Communications Magazine*, Vol. 39, no. 1, January 2001, pp. 136-142.

[48] S. Yao, S. J. B. Yoo, B. Mukherjee, and S. Dixit, "All-Optical Packet Switching for Metropolitan Area Networks: Opportunities and Challenges", *IEEE Communications Magazine*, Vol. 39, no. 3, March 2001, pp. 142-148.

[49] M. Veeraraghavan, R. Karri, T. Moors, M. Karol, and R. Grobler, "Architectures and Protocols that Enable New Applications on Optical Networks", *IEEE Communications Magazine*, Vol. 39, no. 3, March 2001, pp. 118-127.

[50] M. J. O'Mahony, D. Simeonidou, D. K. Hunter, and A. Tzanakaki, "The Application of Optical Packet Switching in Future Communication Networks", *IEEE Communications Magazine*, Vol. 39, no. 3, March 2001, pp. 128-135.

[51] S. Bjørnstad, N. Stol, and D. R. Hjelme, "Quality of Service in Optical Packet Switched DWDM Transport Networks", Proceedings of SPIE Asia-Pacific Optical and Wireless Communications Conference and Exhibition (APOC), Proceedings of SPIE Vol. 4910, pp. 63-74, 2002.

[52] M. Nord, S. Bjørnstad, and C. M. Gauger, "OPS or OBS in the Core Networks?" Proceedings of 7th IFIP Working Conference on Optical Network Design and Modelling (ONDM 2003), Proceedings of the 7th IFIP Working Conference on Optical Network Design and Modelling (ONDM 2003), Budapest, pp. 179-198, 2003.

[53] H. Øverby and N. Stol, "Effects of bursty traffic in service differentiated Optical Packet Switched networks", *Optics Express*, Vol. 12, no. 3, February 2004, pp. 410-415.

[54] J. Turner, "Terabit burst switching," Department of Computer Science, Washington University, St. Louis, Technical Report WUCS-97-49, December 1997.

[55] C. Qiao, "Optical Burst Switching - A Novel Paradigm", Proceedings of Optical Internet Workshop '97, (see related links at http://www.isi.edu/~workshop/oi97/), October, 1997.

[56] M. Yoo and C. Qiao, "A Novel Switching Paradigm for Buffer-Less WDM Networks", Proceedings of Optical Fiber Communication Conference (OFC'99), San Diego, California, pp. 177-179, February, 1999.

[57] 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, January 1999, pp. 69-84.

[58] C. Qiao and M. Yoo, "Choices, features and issues in optical burst switching", *Optical Networks Magazine*, Vol. 1, no. 2, Abril 2000, pp. 36-44.

[59] C. M. Gauger, "Trends in Optical Burst Switching", Proceedings of SPIE ITCOM 2003, Proceedings of SPIE vol. 5247, Orlando, USA, September 7-11, 2003, 2003.

[60] T. Battestilli and H. Perros, "An introduction to Optical Burst Switching", *IEEE Communications Magazine*, Vol. 41, no. 8, August 2003, pp. S10-S15.

[61] G. N. Rouskas and L. Xu, "Optical Packet Switching," in *Optical WDM Networks: Past Lessons and Path Ahead*, K. M. Sivalingam and S. Subramaniam (Eds.), Kluwer Academic Publishers, Norwell, Massachusetts, 2004, pp. 1-19.

[62] L. Xu, "Performance Analysis of Optical Burst Switched Newtorks," *A dissertation submitted to North Carolina State University in partial satisfaction of the requirements for the Degree of Doctor of Philosophy*, Department of Computer Science, Raleigh, 2002.

[63] Y. Chen, C. Qiao, and X. Yu, "Optical Burst Switching: A New Area in Optical Networking Research", *IEEE Network*, Vol. 18, no. 3, May/June 2004, pp. 16-23.

[64] J. P. Jue and V. M. Vokkarane, *Optical Burst Switched Networks*, Springer, New York, USA, 2005.

[65] S. Yao, B. Mukherjee, and S. Dixit, "Advances toward Optical Subwavelength Switching," in *IP Over WDM - Building the Next-Generation Optical Internet*, S. Dixit (Ed.), John Wiley & Sons, Inc., New Jersey, 2003, pp. 157-180.

[66] H. G. Perros, *Connection-oriented Networks: SONET/SDH, ATM, MPLS and Optical Networks*, John Wiley & Sons, West Sussex, England, February 2005.

[67] J. M. H. Elmirghani and E. H. Sargent_(Eds.), "Packet-Oriented Photonic Networks", *IEEE Communications Magazine*, Vol. 40, no. 9, 2002, pp. 56-87.

[68] S. S. Dixit and P. J. Lin_(Eds.), "Advances in Packet Switching/Routing in Optical Networks", *IEEE Communications Magazine*, Vol. 39, no. 2, 2001, pp. 79-113.

[69] F. Callegati, M. Casoni, C. Raffaelli, and B. Bostica, "Packet optical networks for TCP-IP backbones", *IEEE Communications Magazine*, Vol. 37, no. 1, 1999, pp. 124-129.

[70] B. Rajagopalan, J. Luciani, D. Awduche, B. Cain, B. Jamoussi, and D. Saha, "IP over optical networks: A framework," IETF Draft, <draft-ietf-ipo-framework-01.txt> 2002.

[71] D. Awduche and Y. Rekhter, "Multiprotocol label switching: Combining MPLS traffic engineering control with optical cross-connects", *IEEE Communications Magazine*, Vol. 39, no. 3, March 2001, pp. 111-116.

[72] A. Banerjee, J. Drake, J. P. Lang, B. Turner, Kompella, and Y. Rekhter, "Generalized multiprotocol label switching", *IEEE Communications Magazine*, Vol. 39, no. 1, January 2001, pp. 144-150.

[73] M. Yoo and C. Qiao, "Just-Enough-Time (JET): A High Speed Protocol for Bursty Traffic in Optical Networks", Proceedings of IEEE/LEOS Conf. on Technologies For a Global Information Infrastructure, pp. 26-27, August, 1997.

[74] 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, December 2000, pp. 2019-2037.

[75] 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, February 2002, pp. 82-89.

[76] C. S. R. Murthy and M. Gurusamy, *WDM Optical Networks, Concepts, Design and Algorithms*, Prentice Hall PTR, New Jersey, 2002.

[77] N. Ghani, "Lambda-Labeling: A framework for IP-over-WDM using MPLS", *Optical Networks Magazine*, Vol. 1, no. 2, 2000, pp. 45-58.

[78] R. Doverspike and J. Yates, "Challenges for MPLS in optical networks restoration", *IEEE Communications Magazine*, Vol. 39, no. 2, 2001, pp. 89-96.

[79] D. Awduche and Y. Rekhter, "Multiprotocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnects", *IEEE Communications Magazine*, Vol. 39, no. 3, March 2001, pp. 111-116.

[80] J. Teng and G. N. Rouskas, "A Detailed Analisys and Performance Comparison of Wavelength Reservation Schemes for Optical Burst Switched Networks", *Photonic Network Communications*, Vol. 9, no. 3, May 2005, pp. 311-335.

[81] M. Yoo and C. Qiao, "A New Optical Burst Switching Protocol for Supporting Quality of Service", Proceedings of Photonics East - Technical Conference on All-Optical Networking: Architecture, Control, and Management Issues, 3531, SPIE, pp. 396-405, November 3-5, 1998.

[82] A. H. Zaim, I. Baldine, M. Cassada, G. N. Rouskas, H. G. Perros, and D. Stevenson, "The JumpStart Just-In-Time Signaling Protocol: A Formal Description Using EFSM", *Optical Engineering*, Vol. 42, no. 2, February 2003, pp. 568-585.

[83] A. H. Zaim, I. Baldine, M. Cassada, G. N. Rouskas, H. G. Perros, and D. Stevenson, "Formal description of jumpstart (Just-In-Time) signaling protocol using EFSM", Proceedings of Opticomm 2002: Optical Networking and Communications, 4874, Boston, MA, SPIE, pp. 160-173, July 29-31, 2002.

[84] I. Baldine, G. N. Rouskas, H. G. Perros, and D. Stevenson, "Signaling Support for Multicast and QoS within the JumpStart WDM Burst Switching Architecture", *Optical Networks Magazine*, Vol. 4, no. 6, November/December 2003, pp. 68-80.

[85] I. Baldine, H. G. Perros, G. N. Rouskas, and D. Stevenson, "JumpStart: A Just-In-Time Signaling Architecture for Optical Burst-Switched Networks", Proceedings of Networking 2002, Pisa, Italy, pp. 1081-1086, May 19-24, 2002.

[86] "The JumpStart project," 2001, http://jumpstart.anr.mcnc.org, accessed at May 2004.

[87] E. Monteiro and F. Boavida, *Engenharia de Redes Informáticas*, FCA - Editora de Informática, Lisboa, 2000.

[88] "International Organization for Standardization," http://www.iso.org, accessed at June 2004.

[89] P. Loureiro, *TCP/IP em Redes Microsoft para Profissionais*, 4ª ed, FCA - Editora de Informática, Lisboa, 1998.

[90] V. M. Vokkarane, K. Haridoss, and J. P. Jue, "Threshold-Based Burst Assembly Policies for QoS Support in Optical Burst-Switched Networks", Proceedings of SPIE Optical Networking and Communication Conference (OptiComm 2002), Vol. 4874, Boston, MA, pp. 125-136, July, 2002.

[91] C. Kan, H. Balt, S. Michel, and D. Verchere, "Information model of an optical burst edge switch", Proceedings of IEEE International Conference on Communications (ICC 2002), 5, New York, NY, USA, pp. 2717-2721, April 28 - May 2, 2002.

[92] S.-y. Oh, H. H. Hong, and M. Kang, "A Data Burst Assembly Algorithm in Optical Burst Switching Networks", *Electronics and Telecommunications Research Institute Journal*, Vol. 24, no. 4, August 2002, pp. 311-322.

[93] V. Vokkarane and J. Jue, "Prioritized Burst Segmentation and Composite Burst Assembly Techniques for QoS Support in Optical Burst-Switched Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 21, no. 7, September 2003, pp. 1198-1209.

[94] L. Xu, H. G. Perros, and G. N. Rouskas, "A Queueing Network Model of an Edge Optical Burst Switching Node", Proceedings of IEEE INFOCOM, Vol. 3, San Francisco, California, pp. 2019-2029, March 30 - Abril 3, 2003.

[95] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, "Control architecture in optical burst-switched WDM networks", *IEEE Journal on Selected Areas in Communications*, Vol. 18, no. 10, October 2000, pp. 1838-1851.

[96] P. Gambini, M. Renaud, C. Guillemot, F. Callegati, I. Andonovic, B. Bostica, D. Chiaroni, G. Corazza, S. L. Danielsen, P. Gravey, P. B. Hansen, M. Henry, C. Janz, A. Kloch, R. Krähenbühl, C. Raffaelli, M. Schilling, A. Talneau, and L. Zucchelli, "Transparent Optical Packet Switching: Network Architecture and Demonstrators in the KEOPS Project", *IEEE Journal on Selected Areas in Communications*, Vol. 16, no. 7, September 1998, pp. 1245-1259.

[97] D. J. Blumenthal, P. R. Prucnal, and J. R. Sauer, "Photonic packet switches: architectures and experimental implementations", *Proceedings of the IEEE*, Vol. 82, no. 11, November 1994, pp. 1650-1667.

[98] S. L. Danielsen, B. Mikkelsen, C. Joergensen, T. Durhuus, and K. E. Stubkjaer, "WDM Packet Switch Architectures and Analysis of the Influence of Tuneable Wavelength Converters on the Performance", *IEEE Journal of Lightwave Technology*, Vol. 15, no. 2, February 1997, pp. 219-227.

[99] S. L. Danielsen, C. Joergensen, B. Mikkelsen, and K. E. Stubkjaer, "Optical Packet Switched Network Layer Without Optical Buffers", *IEEE Photonics Technology Letters*, Vol. 10, no. 6, June 1998, pp. 896-898.

[100] C. Kan, H. Balt, S. Michel, and D. Verchere, "Network-element view information model for an optical burst core switch", Proceedings of Asia-Pacific Optical and Wireless Communications (APOC), SPIE Vol. 4585, Beijing, China, SPIE, pp. 115-125, September 12-16, 2001.

[101] F. Callegati, H. C. Cankaya, Y. Xiong, and M. Vandenhoute, "Design Issues of Optical IP Routers for Internet Backbone Applications", *IEEE Communications Magazine*, Vol. 37, no. 12, December 1999, pp. 124-128.

[102] A. Ge, F. Callegti, and Lakshman, "On optical burst switching and Self-similar traffic", *IEEE Communications Letters*, Vol. 4, no. 3, March 2000, pp. 98-100.

[103] V. M. Vokkarane, Q. Zhang, J. P. Jue, and B. Chen, "Generalized burst assembly and scheduling techniques for QoS support to optical burst-switched networks", Proceedings of IEEE Globecom 2002, Taipei, Taiwan, November 17-21, 2002.

[104] X. Cao, J. Li, Y. Xen, and C. Qiao, "Assembling TCP/IP Packets in Optical Burst Switched Networks", Proceedings of IEEE Globecom 2002, Taipei, Taiwan, November 17-21, 2002.

[105] X. Yu, Y. Chen, and C. Qiao, "Performance Evaluation of Optical Burst Switching with Assembled Burst Traffic Input", Proceedings of IEEE GLOBECOM 2002, Taipei, Taiwan, November 17-21, 2002.

[106] K. Dolzer, "Assured Horizon - An efficient framework for service differentiation in optical burst switched networks", Proceedings of SIE Optical Networking and Communications Conference (OptiComm 2002), SPIE (Ed.), Vol. 4874, Boston, SPIE, July 30-31, 2002.

[107] K. Dolzer, "Assured Horizon - A new combined framework for burst assembly and reservation in optical burst switched networks", Proceedings of 7th European Conference on Networks and Optical Communications (NOC 2002), R. H. a. D. W. F. (Eds.) (Ed.), Darmstadt, Germany, June 18-21, 2002.

[108] J. Phuritatkul and Y. Ji, "Buffer and Bandwidth allocation Algorithms for Quality of Service Provisioning in WDM Optical Burst Switching Networks", Proceedings of 7th IEEE High Speed Networks and Multimedia Communications (HSNMC 2004), Z. M. a. P. L. (Eds.) (Ed.), Vol. LNCS 3079, Toulouse, France, Springer, pp. 912-920, June 30 - July 02, 2004.

[109] T. Tachibana, T. Ajima, and ShojiKasahara, "Round-RobinBurst Assembly and Constant TransmissionScheduling for Optical Burst Switching Networks", Proceedings of IEEE GLOBECOM 2003, San Francisco, California, IEEE, pp. 2772-2776, December 1-5, 2003.

[110] C. Qiao, "Methods to route and re-route data in OBS/LOBS and other burst switched networks," Patent No. US 2003/0206521 A1, United States, November 6, 2003.

[111] I. Widjaja, "Performance Analysis of Burst Admission Control Protocols", *IEE Proceeding - Communications*, Vol. 142, no. 1, February 1995, pp. 7-14.

[112] A. Detti and M. Listanti, "Application of Tell & Go and Tell & Wait Reservation Strategies in a Optical Burst Switching Network: a Performance Comparison", Proceedings of IEEE International Conference on Telecommunication (ICT), 2, Bucharest, Romania, pp. 540-548, June 4-7, 2001.

[113] L. Xu, H. G. Perros, and G. N. Rouskas, "Access Protocols for Optical Burst-Switched Ring Networks", *Information Sciences*, Vol. 149, no. 1-3, January 2003, pp. 75-81.

[114] M. Düser and P. Bayvel, "Analysis of a dynamically wavelength-routed optical burst switched network architecture", *IEEE Journal of Lightwave Technology*, Vol. 20, no. 4, April 2002, pp. 574-585.

[115] J. Teng, "A Study of Optical Burst Switched Networks with theJumpstart Just In Time Signaling Protocl," *A dissertation submitted to University of North Carolina State University in partial satisfaction of the requirements for the Degree of Doctor of Philosophy*, Department of Computer Science, Raleigh, 2004.

[116] V. Vokkarane and J. Jue, "Burst Segmentation: an Approach for Reducing Packet Loss in Optical Burst-Switched Networks", *Optical Networks Magazine*, Vol. 4, no. 6, November/December 2003, pp. 81-89.

[117] M. Yoo, C. Qiao, and S. Dixit, "A Comparative Study of Contention Resolution Policies in Optical Burst Switched WDM Networks", Proceedings of Terabit Optical Networking: Architecture, Control, and Management Issues 2000 (VV05), Vol. 4213, Boston, MA, USA, SPIE, pp. 124-135, November 6-7, 2000.

[118] Z. Haas, "The "Staggering Switch": An electronically Controlled Optical Packet Switch", *IEEE Journal of Lightwave Technology*, Vol. 11, no. 5/6, May/June 1993, pp. 925-936.

[119] A. S. Acampora and I. A. Shah, "Multihop Lightwave Networks: A Comparison of Store-and-Forward and Hot-Potato Routing", *IEEE Transactions on Communications*, Vol. 40, no. 6, June 1992, pp. 1082-1090.

[120] F. Forghieri, A. Bononi, and P. R. Prucnal, "Analysis and Comparison of Hot-Potato and Single-Buffer Deflection Routing in Very High Bit Rate Optical Mesh Networks", *IEEE Transactions on Communications*, Vol. 43, no. 1, January 1995, pp. 88-98.

[121] B. Ramamurthy and B. Mukherjee, "Wavelength Conversion in WDM Networking", *IEEE Journal on Selected Areas in Communications*, Vol. 16, no. 7, September 1998, pp. 1061-1073.

[122] A. Bononi, G. A. Castañón, and O. K. Tonguz, "Analysis of Hot-Potato Optical Networks with Wavelength Conversion", *IEEE Journal of Lightwave Technology*, Vol. 17, no. 4, April 1999, pp. 525-534.

[123] X. Wang, H. Morikawa, and T. Aoyama, "Burst optical deflection routing protocol for wavelength routing wdm networks", Proceedings of Opticomm, Dallas, Texas, USA, SPIE, pp. 257-266, 2000.

[124] Y. Chen, H. Wu, D. Xu, and C. Qiao, "Performance Analysis of Optical Burst Switched Node with Deflection Routing", Proceedings of IEEE International Conference on Communications (ICC 2003), Vol. 2, Anchorage, Alaska, USA, IEEE, pp. 1355-1359, May, 11-15, 2003.

[125] X. Wang, H. Morikawa, and T. Aoyama, "Deflection routing protocol for burst switching WDM mesh networks", Proceedings of SPIE/IEEE Terabit Optical Networking: Architecture, Control, and Management Issues, Boston, USA, SPIE, pp. 242-252, November, 2000.

[126] V. M. Vokkarane, J. P. Jue, and S. Sitaraman, "Burst Segmentation: An Approach For Reducing Packet Loss In Optical Burst Switched Networks", Proceedings of IEEE International Conference on Communications (ICC 2002), 5, New York, NY, IEEE, pp. 2673-2677, April, 2002.

[127] V. M. Vokkarane and J. P. Jue, "Prioritized Routing and Burst Segmentation for QoS in Optical Burst- Switched Networks", Proceedings of Optical Fiber Communication Conference and Exhibit (OFC 2002), Anaheim, CA, USA, pp. 221-222, March 17-22, 2002.

[128] V. Vokkarane and J. Jue, "Segmentation-Based Non-Preemptive Scheduling Algorithms for Optical Burst-Switched Networks", Proceedings of The First

International Workshop on Optical Burst Switching, Dallas, Texas, USA, October 16, 2003.

[129] C. M. Gauger, "Contention Resolution in Optical Burst Switching Networks," in *Contention resolution in optical burst switching networks Advanced Infrastructures for Photonic Networks: WG 2 Intermediate Report*, 2002, pp. 62-82.

[130] C. M. Gauger, "Performance of Converter Pools for Contention Resolution in Optical Burst Switching Networks", Proceedings of Opticomm 2002: Optical Networking and Communications, N. Ghani and K. M. Sivalingam (Eds.), Vol. 4874, SPIE, pp. 109-117, 2002.

[131] A. Maach and G. v. Bochmann, "Segmented Burst Switching: Enhancement of Optical Burst Switching to decrease loss rate and support quality of service", Proceedings of Optical Network Design and Modeling (ONDM 2002), Torino, Italy, pp. 69-84, February, 4-6, 2002.

[132] A. Detti, V. Eramo, and M. Listanti, "Optical burst switching with burst drop(obs/bd): an easy obs improvement", Proceedings of IEEE International Conference Communications (ICC 2002), 5, New York, NY, USA, IEEE, pp. 2687-2691, April 28 - May 2, 2002.

[133] J. Teng and G. N. Rouskas, "A Comparison of the JIT, JET, and Horizon Wavelength Reservation Schemes on A Single OBS Node", Proceedings of The First International Workshop on Optical Burst Switching (WOBS), Dallas, Texas, USA, October 13-18, 2003.

[134] T. J. Schriber and R. W. Andrews, "Interactive analysis of simulation output by the method of batch means", Proceedings of 11th conference on Winter simulation, 2, San Diego, CA, United States, IEEE Press, pp. 513-526, 1979.

[135] H. Perros, "Computer Simulation Techniques: The definitive introduction!," 2003, http://www.csc.ncsu.edu/faculty/perros/, accessed at December 18th, 2003.

[136] M. Yoo, C. Qiao, and S. Dixit, "QoS Performance of Optical Burst Switching in IP-Over-WDM Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 18, no. 10, October 2000, pp. 2062-2071.

[137] L. Xu, H. G. Perros, and G. N. Rouskas, "Performance Analysis of an Edge Optical Burst Switching Node with A Large Number of Wavelengths", Proceedings of 18th International Teletraffic Congress (ITC-18), Berlin, Germany, pp. 891-900, August 31-September 5, 2003.

[138] T. Battestilli and H. Perros, "End-to-End Burst Loss Probabilities in an OBS Network with Simultaneous Link Possession", Proceedings of The Third International Workshop on Optical Burst Switching (WOBS 2004), San José, CA, USA, October, 25th, 2004.

[139] C. G. Boncelet and D. L. Mills, "A labeling algorithm for Just-In-Time scheduling in TDMA networks", Proceedings of Conference on Communications Architecture & Protocols - ACM SIGCOMM, Baltimore, Maryland, USA, ACM Press, pp. 170-175, August 17-20, 1992.

[140] Y. Zhu, G. N. Rouskas, and H. G. Perros, "A Comparison of Allocation Policies in Wavelength Routing Networks", *Photonic Networks Communication Journal*, Vol. 2, no. 3, August 2000, pp. 265-293.

[141] I. Baldine, M. Cassada, A. Bragg, G. Karmous-Edwards, and D. Stevenson, "Just-in-Time Optical Burst Switching Implementation in the ATDnet All-Optical Networking Testbed", Proceedings of IEEE GLOBECOM 2003, IEEE (Ed.), Vol. 5, San Francisco, USA, IEEE, pp. 2777-2781, December 1-5, 2003.

[142] L. McAdams, I. Richer, and S. Zabele, "TBONE: TestBed for all-Optical NEtworking", Proceedings of IEEE/LEOS Summer Topical Meetings, pp. 3_13, 3_14, 1994.

[143] I. Baldine, A. Bragg, G. Evans, M. Pratt, M. Singhai, D. Stevenson, and R. Uppalli, "JumpStart Deployments in Ultra High Performance Optical Networking Testbeds", *IEEE Communications Magazine*, Vol. 43, no. 11, November 2005, pp. S18-S25.

[144] P. Bartford and L. Landweber, "Bench-style Network Research in an Internet Instance Laboratory", *ACM SIGCOMM Computer Communications Review*, Vol. 33, no. 3, Julho 2003, pp. 21.

[145] D. E. Knuth, *The Art of Computer Programming*, 2nd ed, Addison-Wesley Publishing Company, Reading, Massachusetts, 1980.

[146] V. Paxson and S. Floyd, "Wide Area Traffic: The Failure of Poisson Modeling", *IEEE Transactions on Networking*, Vol. 3, no. 3, 1995, pp. 226-244.

[147] A. Schäefer, "Self-Similar Network Traffic," 2003, http://goethe.ira.uka.de/~andreas/Research/Fractal_Traffic/Fractal_Traffic.html, accessed at February 3rd, 2004.

[148] Sun Microsystems Inc., "Class Math," 2003, http://java.sun.com/j2se/1.4.2/docs/api/java/lang/Math.html/, accessed at February 23th, 2004.

[149] S. Ma and C. Ji, "Modeling heterogeneous network traffic in wavelet domain", *IEEE / ACM Transactions on Networking*, Vol. 9, no. 5, 2001, pp. 634-649.

[150] 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, 1997, pp. 71-86.

[151] M. Sridharan, M. V. Salapaka, and A. K. Somani, "A Practical Approach to Operating Survivable WDM Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 20, no. 1, January 2002, pp. 34-46.

[152] S. Ramesh, G. N. Rouskas, and H. G. Perros, "Computing blocking probabilities in multiclass wavelength-routing networks with multicast calls", *IEEE Journal on Selected Areas in Communications*, Vol. 20, no. 1, January 2002, pp. 89-96.

[153] T. K. Nayak and K. N. Sivarajan, "A New Approach to Dimensioning Optical Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 20, no. 1, January 2002 2002, pp. 134-148.

[154] M. J. O'Mahony, "Results from the COST 239 Project: Ultra-high Capacity Optical Transmission Networks", Proceedings of 22nd European Conference on Optical Communication (ECOC'96), Vol. 2, Oslo, Norway, pp. 2.11-2.18, September 15-19, 1996.

[155] B. W. Arden and H. Lee, "Analysis of Chordal Ring Networks", *IEEE Transactions on Computers*, Vol. C-30, no. 4, 1981, pp. 291-295.

[156] FCCN, "Fundação para a Computação Científica Nacional," 2004, http://www.fccn.pt, accessed at September, 15th.

[157] "REFER Telecom SA," 2004, http://www.refertelecom.pt, accessed at September, 15th.

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

[159] L. Xu, H. G. Perros, and G. N. Rouskas, "A Simulation Study of Access Protocols for Optical Burst-Switched Ring Networks", Proceedings of Networking 2002, Pisa, Italy, pp. 863-874, May 19-24, 2002.

[160] B.-C. Kim, Y.-Z. Cho, J.-H. Lee, Y.-S. Choi, and D. Montgomery, "Performance of Optical Burst Switching Techniques in Multi-Hop Networks", Proceedings of IEEE Globecom 2002, Taipei, Taiwan, IEEE, 17- 27 November, 2002.

[161] A. Zalesky, H. L. Vu, Z. Rosberg, E. Wong, and M. Zukerman, "Modelling and Performance Evaluation of Optical Burst Switched Networks with Deflection Routing and Wavelength Reservation", Proceedings of IEEE Infocom 2004, Hong Kong, IEEE, March 7-11, 2004.

[162] D. Lee, L. Libman, and A. Orda, "Path Protection and Blocking Probability Minimization in Optical Networks", Proceedings of IEEE Infocom 2004, Hong Kong, IEEE, March 7-11, 2004.

[163] P. B. Chu, S.-S. Lee, and S. Park, "MEMS: The Path to Large Optical Crossconnects", *IEEE Communications Magazine*, Vol. 40, no. 3, March 2002, pp. 80-87.

[164] H. J. Chao and S. Y. Liew, "A new optical cell switching paradigm", Proceedings of First International Workshop on Optical Burst Switching, Dallas, Texas, USA, October 13-18, 2003.

[165] E. Modiano, "Optical Flow Switching in the Next Generation Internet", Proceedings of Gigabit Networking Workshop (GBN 2000), Tel Aviv, Israel, 26 March 2000, 2000.

[166] J. J. P. C. Rodrigues, M. M. Freire, and P. Lorenz, "Performance Implications of Meshing Degree for Optical Burst Switched Networks Using One-way Resource Reservation Protocols", *Telecommunications Systems*, Vol. 30, no. 1-3, November 2005, pp. 35-47.

[167] E. F. Haselton, "A PCM frame switching concept leading to burst switching network architecture", *IEEE Communications Magazine*, Vol. 21, no. 6, September 1983, pp. 13-19.

[168] S. R. Amstutz, "Burst Switching - An Update", *IEEE Communications Magazine*, Vol. 27, no. 9, September 1989, pp. 50-57.

[169] S. R. Amstutz, "Burst Switching - An Introduction", *IEEE Communications Magazine*, Vol. 21, no. 8, November 1983, pp. 36-42.

[170] D. L. Mills, C. G. Boncelet, J. G. Elias, P. A. Schragger, and A. W. Jackson, "Highball: a high speed, reserved-access, wide-area network," Electrical Engineering Department, University of Delaware Technical Report 90-9-3, September 1990.

[171] H. Perros, *An introduction to ATM networks*, Wiley, New York, 2001.

[172] C. Qiao, "Labeled Optical Burst Switching for IP-over-WDM Integration", *IEEE Communications Magazine*, Vol. 38, no. 9, Sptember 2000, pp. 104-114.

[173] A. Kaheel and H. Alnuweiri, "Quantitative QoS Guarantees in Labeled Optical Burst Switching Networks", Proceedings of IEEE Global Communications Conference (GLOBECOM 2004), Dallas, TX, IEEE, November 29 - December 03, 2004.

[174] A. Detti, V. Eramo, and M. Listanti, "Performance evaluation of a new technique for IP support in a WDM optical network: optical composite burst switching (OCBS)", *IEEE Journal on Selected Areas in Communications*, Vol. 20, no. 2, February 2002, pp. 154-165.

[175] A. Bragg, I. Baldine, and D. Stevenson, "A transport layer architectural framework for optical burst switched (OBS) networks", Proceedings of 1st International Workshop on Optical Burst Switching, Dallas, Texas, 2003.

[176] M. Rose and K. McCloghrie, *Structure and Identification of Management Information for TCP/IP-based Internets*, RFC 1155, http://www.ietf.org/rfc/rfc1155.txt?number=1155, May 1990.

[177] T. Socolofsky and C. Kale, *A TCP/IP Tutorial*, RFC 1180, http://www.ietf.org/rfc/rfc1180.txt?number=1180, January 1991.

[178] Y. Pouffary and A. Young, *ISO Transport Service on top of TCP (ITOT)*, RFC 2126, http://www.ietf.org/rfc/rfc2126.txt?number=2126, March 1997.

[179] G. Kessler and S. Shepard, *A Primer On Internet and TCP/IP Tools and Utilities*, RFC 2151, http://www.ietf.org/rfc/rfc2151.txt?number=2151, June 1997.

[180] J. White, M. Zukerman, and H. L. Vu, "A Framework for Optical Burst Switching Network Design", *IEEE Communications Letters*, Vol. 6, no. 6, June 2002, pp. 268-270.

[181] M. Yoo and Y. Kim, "Internetworking Optical Internet and Optical Burst Switching," in *IP Over WDM - Building the Next-Generation Optical Internet*, S. S. Dixit (Ed.), John Wiley and Sons, Inc., New Jersey, 2003, pp. 397-420.

[182] F. Farahmand, V. M. Vokkarane, and J. P. Jue, "Practical Priority Contention Resolution for Slotted Optical Burst Switching Networks", Proceedings of The First International Workshop on Optical Burst Switching (WOBS), Dallas, Texas, USA, October 13-18, 2003.

[183] M. Aldwairi, M. Guled, M. Cassada, M. Pratt, D. Stevenson, and P. Franzon, "Switch Arqchitecture for Optical Burst Switching Networks", Proceedings of First International Workshop on Optical Burst Switching (WOBS), Dallas, Texas, USA, October 13-18, 2003.

[184] J. Y. Choi, H. L. Vu, C. W. Cameron, M. Zukerman, and M. Kang, "The effect of Burst Assembly on Performance of Optical Burst Switched Networks", Proceedings of The International Conference on Information Networking (ICOIN 2004), III, Busan, Korea, Springer, pp. 1197-1206, February 18-20, 2004.

[185] J. Turner, "Terabit Burst Switching," 1997, http://www.arl.wustl.edu/projects/archive/burst/, accessed at August 17th, 2004.

[186] C. Gauger, K. Dolzer, J. Späth, and S. Bodamer, "Service Differentiation in Optical Burst Switching Networks", Proceedings of Beiträge zur 2. ITG Fachtagung Photonische Netze, Dresden, Germany, pp. 124-132, March 12-13, 2001.

[187] J. I. Dadap, P. B. Chu, I. Brener, C. Pu, C. D. Lee, K. Bergman, N. Bonadeo, T. Chau, M. Chou, R. Doran, R. Gibson, R. Harel, J. J. Johnson, S. S. Lee, S. Park, D. R. Peale, R. Rodriguez, D. Tong, M. Tsai, C. Wu, W. Zhong, E. L. Goldstein, L. Y. Lin, and J. A. Walker, "Modular MEMS-Based Optical Cross-Connect With Large Port-Count", *IEEE Photonics Technology Letters*, Vol. 15, no. 12, December 2003, pp. 1773-1775.

[188] R. E. Wagner, R. C. Alferness, A. A. M. Saleh, and M. S. Goodman, "MONET: Multiwavelength optical networking", *IEEE Journal of Lightwave Technology*, Vol. 14, no. 6, June 1996, pp. 1349-1355.

[189] L. D. Garrett *et al.*, "The MONET New Jersey network demonstration", *IEEE Journal on Selected Areas in Communications*, Vol. 16, no. 7, September 2000, pp. 1199 - 1219.

[190] W. T. Anderson *et al.*, "The MONET project-a final report", *IEEE Journal of Lightwave Technology*, Vol. 18, no. 12, December 2000, pp. 1988 - 2009.

[191] "ARDA - Advanced Research and Development Agency," http://www.ic-arda.org, accessed at August 25, 2003.

[192] "MCNC Research and Development Institute," http://www.mcnc.org, accessed at July 21, 2004.

[193] J. Xu, C. Qiao, J. Li, and G. Xu, "Efficient Channel Scheduling Algorithms in Optical Burst Switched Networks", Proceedings of IEEE INFOCOM 2003, San Francisco, IEEE, March 30 - April 03, 2003.

[194] Y. Xiong, M. Vandenhoute, and H. C. Cankaya, "Design and analysis of optical burst-switched networks", Proceedings of SPIE'99 All-Optical Networking 1999:

Architecture, Control, and Management Issues, C. Q. John M. Senior, Sudhir Dixit; Eds (Ed.), 3843, Proceedings of SPIE, pp. 112-119, August, 1999.

[195] M. Yoo, C. Qiao, and S. Dixit, "Optical Burst Switching for Service Differentiation in the next Generation Optical Internet", *IEEE Communications Magazine*, Vol. 39, no. 2, February 2001, pp. 98-104.

[196] M. Yang, S. Q. Zheng, and D. Verchere, "A QoS Supporting Scheduling Algorithm for Optical Burst Switching DWDM Networks", Proceedings of IEEE GLOBECOM 2001, 1, San Antonio, Texas, USA, IEEE, pp. 86-91, November 25-29, 2001.

[197] D. K. Hunter, W. D. Cornwell, T. H. Gilfedder, A. Franzen, and I. Andonovic, "SLOB: A Switch with Large Optical Buffers for Packet Switching", *IEEE Journal of Lightwave Technology*, Vol. 16, no. 10, October 1998, pp. 1725-1736.

[198] S. Subrarnaniam, M. Azizoglu, and A. K. Somani, "All-Optical Networks with Sparse Wavelength Conversion", *IEEE / ACM Transactions on Networking*, Vol. 4, no. 4, August 1996, pp. 544-557.

[199] C.-F. Hsu, T.-L. Liu, and N.-F. Huang, "Performance analysis of deflection routing in optical burst-switched networks", Proceedings of IEEE INFOCOMM 2002, 1, New York, USA, IEEE, pp. 66-73, June 23-27, 2002.

[200] A. Kaheel, T. Khattab, A. Mohamed, and H. Alnuweiri, "Quality-of-Service Mechanisms in IP-over-WDM Networks", *IEEE Communications Magazine*, Vol. 40, no. 12, December 2002, pp. 38-43.

[201] X. Xiao and L. M. Ni, "Internet QoS: A Big Picture", *IEEE Network*, Vol. 13, no. 2, March-April 1999, pp. 8-18.

[202] B. Wydrowski and M. Zukerman, "QoS in Best-Effort Networks", *IEEE Communications Magazine*, Vol. 40, no. 12, December 2002, pp. 44-49.

[203] R. Braden, D. Clark, and S. Shenker, *Integrated Services in the Internet Architecture: an Overview*, RFC 1633, http://www.ietf.org/rfc/rfc1633.txt, June 1994.

[204] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *An Architecture for Differentiated Services*, RFC 2475, http://www.ietf.org/rfc/rfc2475.txt, December 1998.

[205] K. Nichols, S. Blake, F. Baker, and D. Black, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, RFC 2474, http://www.ietf.org/rfc/rfc2474.txt, December 1998.

[206] Q. Zhang, V. M. Vokkarane, J. P. Jue, and B. Chen, "Absolute QoS Differentiation in Optical Burst-Switched Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 22, no. 9, November 2004, pp. 1781-1795.

[207] Q. Zhang, V. M. Vokkarane, B. Chen, and J. P. Jue, "Path Clustering: An Approach to Implement Absolute QoS Differentiation in Optical Burst-Switched Networks", Proceedings of IEEE GLOBECOM 2004, Dallas, TX, IEEE, November 29 - December 3, 2004.

[208] M. H. Phùng, K. C. Chua, G. Mohan, M. Motani, and T. C. Wong, "Absolute QoS Signalling and Reservation in Optical Burst-Switched Networks", Proceedings of IEEE Global Communications (GLOBECOM 2004), Dallas, TX, IEEE, November 29 - December 03, 2004.

[209] F. Poppe, K. leavens, H. Michiel, and S. Molenaar, "Quality-of-service differentiation and fairness in optical burst-switched networks", Proceedings of Optical Networking and Communication Conference (OptiComm) 2002, Vol. 4874, Boston, MA, SPIE, July 30-31, 2002.

[210] Y. Chen, M. Hamdi, and D. H. K. Tsang, "Proportional QoS over OBS networks", Proceedings of IEEE Global Communications Conference (GLOBECOM 2001), Vol. 3, San Antonio, TX, IEEE, pp. 1510-1514, November 25-29, 2001.

[211] C. H. Loi, W. Liao, and D. N. Yang, "Service Differentiation in Optical Burst Switched Networks", Proceedings of IEEE Global Communications Conference (GLOBECOM 2002), Vol. 3, Taipei, Taiwan, IEEE, pp. 2313-2317, November, 2002.

[212] 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", Proceedings of IEEE Global Communications Conference (GLOBECOM 2003), San Francisco, USA, IEEE, December, 1-5, 2003.

[213] X. Yu, C. Qiao, and Y. Liu, "TCP Implementations and False Time Out Detection in OBS Networks", Proceedings of The 23rd Conference of the IEEE Communications Society (IEEE INFOCOM 2004), Hong-Kong, IEEE, March 7-11, 2004.

[214] A. Detti and M. Listanti, "Impact of Segments Aggregation on TCP Reno Flows in Optical Burst Switching Networks", Proceedings of IEEE INFOCOM 2002, New York, USA, IEEE, June 23-27, 2002.

[215] S. Gowda, R. K. Shenai, K. M. Sivalingam, and H. C. Cankaya, "Performance Evaluation of TCP over Optical Burst-Switched (OBS) WDM Networks", Proceedings of IEEE ICC, Anchorage, Alaska, 2003.

[216] Q. Zhang, V. M. Vokkarane, Y. Wang, and J. P. Jue, *TCP over Optical Burst-Switched Networks with Optical Burst Retransmission*, Submitted, December, 2004.

[217] L. H. Sahasrabuddhe and B. Mukherjee, "Light-Trees: 0ptical Multicasting for Improved Performance in Wavelength-Routed Networks", *IEEE Communications Magazine*, Vol. 37, no. 2, February 1999, pp. 67-73.

[218] C. Qiao, M. Jeong, A. Guha, X. Zhang, and J. Wei, "WDM Multicasting in IP over WDM Networks", Proceedings of IEEE International Conference on Network Protocols (ICNP'99), Toronto, Canada, pp. 89-96, Oct. 31 - Nov. 03, 1999.

[219] M. Jeong, Y. Xiong, H. C. Cankaya, M. Vandenhoute, and C. Qiao, "Efficient Multicast Schemes for Optical Burst-Switched WDM Networks", Proceedings of IEEE International Conference on Communications (ICC 2000), New Orleans, USA, IEEE, pp. 1289-1294, 2000.

[220] M. Jeong, C. Qiao, Y. Xiong, and M. Vandenhoute, "Bandwidth-efficient dynamic tree-shared multicast in optical burst-switched networks", Proceedings of IEEE International Conference on Communications (ICC 2001), Vol. 2, Helsinki, Finland, IEEE, pp. 630-636, June 11-15, 2001.

[221] M. Jeong, C. Qiao, and Y. Xiong, "Reliable WDM Multicast in Optical Burst-Switched Networks", Proceedings of SPIE Optical Networking and Communications Conference (OptiComm 2000), Dallas, USA, SPIE, pp. 153-166, October, 2000.

[222] S. Sheeshia and C. Qiao, "Burst Grooming in Optical-Burst-Switched Networks", Proceedings of IEEE/SPIE First Workshop on Traffic Grooming in WDM Networks, San José, USA, October 29, 2004.

[223] F. Farahmand, Q. Zhang, and J. P. Jue, *Dynamic Traffic Grooming in Optical Burst-Switched Networks*, Submitted, December, 2004.

[224] A. Banerjee, N. Singhal, J. Zhang, D. Ghosal, C.-N. Chuah, and B. Mukherjee, "A Time-Path Scheduling Problem (TPSP) for Aggregating Large Data Files from Distributed Databases using an OBS Network", Proceedings of IEEE International Conference on Communications (ICC'04), Paris, France, IEEE, June, 20-24, 2004.

[225] E. v. Breusegem, M. d. L. enheer, J. Cheyns, P. Demeester, D. Simeonidou, M. J. O'Mahoney, R. Nejabati, A. Tzanakaki, and I. Tomkos, "An OBS Architecture for Pervasive Grid Computing", Proceedings of The Third International Workshop on Optical Burst Switching (WOBS 2004), San José, USA, October, 25, 2004.

[226] M. D. Leenheer, E. V. Breusegem, P. Thysebaert, B. Volckaert, F. D. Turck, B. Dhoedt, P. Demeester, D. Simeonidou, M. J. O'Mahoney, R. Nejabati, A. Tzanakaki, and I. Tomkos, "An OBS-based Grid Architecture", Proceedings of 2004 IEEE Global Communications Conference Workshops (GLOBECOM 2004), Dallas, TX, USA, IEEE Communications Society, pp. 390-394, November 29 - December 03, 2004.

[227] S. Arima, T. Tachibanay, and S. Kasahara, "FEC-Based Burst Loss Recovery for Multiple-Bursts Transmission in Optical Burst Switching Networks", (submitted), 2005.

[228] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, *Simulation of Communication Systems*, Plenum Press, New York, 1992.

[229] M. M. Freire, "Estudos de Desempenho de Redes de Comunicação Incorporando Aplicações Não Lineares de Amplificadores Ópticos Semicondutores," *A dissertation submitted to Universidade da Beira Interior in partial satisfaction of the requirements for the Degree of Doctor of Philosophy*, Covilhã, 2000.

[230] Z. Rosberg, H. L. Vu, M. Zukerman, and J. White, "Performance analyses of Optical Burst-Switching Networks", *IEEE Journal on Selected Areas in Communications*, Vol. 21, no. 7, September 2003, pp. 1187-1197.

[231] MathWorks, "MATLAB," 1994-2005, http://www.mathworks.com/products/matlab/, accessed at July 23th, 2005.

[232] Center for Hybrid and Embedded Software Systems (CHESS) University of California at Berkeley, "Ptolemy Project," 2005, http://ptolemy.eecs.berkeley.edu/, accessed at July 28th, 2005.

[233] E. A. Lee, "Absolutely Positively On Time: What Would It Take?" *IEEE Computer*, Vol. 38, no. 7, July 2005, pp. 85-87.

[234] "The SSFNet project," http://www.ssfnet.org, accessed at July 2004.

[235] D. Nicol, J. Liu, M. Liljenstam, and G. Yan, "Simulation of large-scale networks using ssf", Proceedings of 2003 Winter Simulation Conference, New Orleans, Louisiana, USA, pp. 650-657, December 7-10, 2003.

[236] CACI Products Company, "About SIMSCRIPT II.5," 2005, http://www.simprocess.com/products/simscript.cfm, accessed at July 26th, 2005.

[237] MathWorks, "Simulink," 1994-2005, http://www.mathworks.com/products/simulink/, accessed at July 23th, 2005.

[238] RSoft Design Group, Inc., "System simulation," 2002, http://www.rsoftdesign.com/products/system_simulation/, accessed at May 2003.

[239] RSoft Design Group, Inc., "LinkSim," http://www.rsoftinc.com/linksim.htm, accessed at May 2003.

[240] OPNET, "OPNET Modeler," http://www.opnet.com/products/modeler/home.html, accessed at June 2004.

[241] "The network simulator ns-2," 2002, http://www.isi.edu/nsnam/ns/, accessed at November, 2002.

[242] "REAL network Simulator," 2002, http://www.cs.cornell.edu/home/skeshav/real/overview.html, accessed at November, 2002.

[243] N. M. Bhide and K. M. Sivalingam, "Design of OWns: Optical Wavelength Division Multiplexing (WDM) Network Simulator", Proceedings of First SPIE Optical Networking Workshop, Dallas, TX, January, 2000.

[244] "The Network Simulator ns-2: Documentation," http://www.isi.edu/nsnam/ns/ns-documentation.html, accessed at November, 2002.

[245] B. Wen, N. M. Bhide, R. K. Shenai, and K. M. Sivalingam, "Optical Wavelength Division Multiplexing (WDM) Network Simulator (OWns): Architecture and Performance Studies", *SPIE Optical Networks Magazine*, Vol. Special Issue on "Simulation, CAD, and Measurement of Optical Networks, no., September/October 2001, pp. 16-26.

[246] Institute of Communication Networks and Computer Engineering, "IND Simulation Library," 2004, http://www.ikr.uni-stuttgart.de/INDSimLib/, accessed at February 17th, 2004.

[247] DAWN Networking Research Lab, "DAWN Research Lab," 2004, http://dawn.cs.umbc.edu/, accessed at January 15th, 2004.

[248] Optical_Internet_Research_Center, "OIRC Optical burst switching Simulator," 2005, http://wine.icu.ac.kr/~obsns/, accessed at July 25th, 2005.

[249] SimReal, "NCTUns Simulator," http://nsl10.csie.nctu.edu.tw/, accessed at June 2005.

[250] R. M. F. Coelho, J. J. P. C. Rodrigues, and M. M. Freire, "Performance Assessment of Wavelength Routed Optical Networks with Shortest Path Routing over Degree Three Topologies", Proceedings of IEEE International Conference on Networks (ICON'2002), Singapore, pp. 3-8, August, 27-30, 2002.

[251] M. M. Freire, J. J. P. C. Rodrigues, and R. M. F. Coelho, "The Role of Network Topologies in the Optical Core of IP-over-WDM Networks with Static Wavelength Routing", *Telecommunications Systems*, Vol. 24, no. 2/4, 2003, pp. 111-122.

[252] M. J. O'Mahony, "A European optical network: design considerations", Proceedings of IEE Colloquium on Transparent Optical Networks: Applications, Architectures and Technology, London, UK, IEE, pp. 1/1-1/6, April 22, 1994.

[253] M. J. O'Mahony, D. Simeonidou, A. Yu, and J. Zhou, "The Design of a European Optical Network", *IEEE Journal of Lightwave Technology*, Vol. 13, no. 5, May 1995, pp. 817-828.

[254] "REFER EP," 2004, http://www.refer.pt, accessed at September 16th.

[255] A. V. Goldberg and R. E. Tarjan, "Expected Performance of Dijkstra's Shortest Path Algorithm", no., June 1996

[256] 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.), Kluver Academic Publishers, 2000.

[257] Y. Alavi, G. Chartrand, L. Lesniak, D. R. Lick, and C. E. Wall, *Graph Theory with Applications to Algorithms and Computer Science*, John Wiley & Sons, New York, 1985.