UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Mechanisms to Reduce Routing Information Inaccuracy Effects: Application to MPLS and WDM Networks

Author: Xavier Masip Bruin

Advisor: Dr. Josep Solé-Pareta
Co-Advisor: Prof. Jordi Domingo-Pascual

COMPUTER ARCHITECTURE DEPARTMENT
TECHNICAL UNIVERSITY OF CATALONIA

A THESIS PRESENTED TO THE UNIVERSITAT POLITECNICA DE CATALUNYA
IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor en Enginyeria de Telecomunicació

Barcelona, June 2003

Dr.	
President	

Dr.	
Secretari	

Dr.	
Vocal	

Dr.	
Vocal	

Dr.	
Vocal	

Data de la defensa pública	
Qualificació	

“No s’ha de treballar ni per avui ni per demà, s’ha de treballar per sempre”

Manuel Hugue

“I never think about the future; it comes too fast”

Albert Einstein

Agraïments

Es molt difícil, quasi bé impossible, incloure totes les persones que d'una manera o altra han pres part en l'elaboració d'aquest document. Tot i això, una manera simple, justa i crec jo força acurada és analitzar l'evolució en els anys. Un primer agraïment ha de ser per els meus pares, qui fa ja molts anys em varen permetre començar i continuar els meus estudis.

En aquest punt, el següent pas és donar les gràcies a totes les persones que han contribuït en aquesta Tesi. En aquí he d'incloure els membres del grup de recerca de xarxes del DAC del qual en formo part, els membres del DAC a Vilanova i especialment els estudiants de la EPSVG en Sergi Benach i en Ivan Colomer.

Explícitament vull agrair a En Josep Solé i En Jordi Domingo el suport i la confiança que com a Directors de la Tesi m'han donat. És molt important l'ajut que en moments difícils, quan malauradament les coses no van com seria desitjable, hom pot rebre de qui t'està dirigint.

Com no, un reconeixement per els revisors anònims que han fet possible la "verificació per publicació" dels diferents resultats que configuren aquesta Tesi.

Aquesta Tesi es va començar ja fa uns quants anys a partir d'un tema de treball en el qual hi estàvem treballant dos persones. Finalment, tant en Sergi com jo hem aconseguit acabar el que 14 anys enrera no ens havíem ni imaginat. Tot i que són ja molts anys treballant plegats, Sergi, és un plaer compartir despatx amb tu.

Fins aquí l'evolució cronològica. Només em queda donar les gràcies a una persona. Sense tu, Carmen, això no s'hagués acabat mai. El teu suport ha sigut vital,

imprescindible i necessari per aconseguir aquest objectiu, tal i com ho ha estat per la resta de coses fonamentals de la meva vida.

Finalment, tan sols complaure'm per l'oportunitat que em dona el fet d'escriure aquests agraïments de poder emprar la llengua del meu país per expressar-me.

Resum

Les xarxes IP tradicionals utilitzen el model de transmissió "best-effort" per transportar tràfic entre clients de la xarxa. Aquest model de transmissió de tràfic no és el més adequat per les aplicacions en temps real com per exemple, vídeo sota demanda, conferències multimedia o realitat virtual que per altra banda tenen cada cop més adeptes entre els clients de la xarxa. A fi de garantir el correcte funcionament d'aquest tipus d'aplicacions, l'estructura de la xarxa ha de ser substancialment modificada amb l'objectiu final de poder optimitzar els seus propis recursos i així poder fer front a aquells tipus de tràfics i de clients que requereixen certes garanties de "Qualitat de Servei" (QoS) per a la seva correcta transmissió.

Aquestes modificacions o millores de la xarxa poden ser perfectament realitzades sota l'entorn d'Enginyeria de Tràfic (Traffic Engineering, TE). Dos són els principals aspectes relacionats amb el funcionament de la xarxa en aquest entorn de TE: els mecanismes de commutació i els mecanismes d'encaminament. Així, per una banda es necessita un mecanisme de commutació molt ràpid en els nodes interns de la xarxa a fi de que els paquets de dades puguin ser processats amb el menor temps possible. En xarxes IP aquest objectiu s'aconsegueix amb el Multiprotocol Label Switching (MPLS). Per altra banda, a fi de garantir certa QoS, les decisions d'encaminament s'han de realitzar tenint en compte quines són les restriccions de QoS sol·licitades per el node client que origina el tràfic. Aquest objectiu s'aconsegueix modificant els esquemes d'encaminament tradicionals, incorporant-hi els paràmetres de QoS en les decisions d'encaminament, generant el que es coneix com algorismes d'encaminament amb QoS (QoS routing).

Centrant-nos en aquest darrer aspecte, la majoria dels algorismes d'encaminament amb QoS existents, realitzen la selecció de la ruta a partir de la informació d'estat de l'enllaç emmagatzemada en les bases de dades d'estat de l'enllaç contingudes en els nodes. Per poder garantir que els successius canvis en

l'estat de la xarxa estiguin perfectament reflectits en aquesta informació d'encaminament, el protocol d'encaminament ha d'incloure un mecanisme d'actualització que faci possible garantir que la selecció de les rutes es fa a partir d'informació acurada de l'estat real de la xarxa. En un entorn IP tradicional, el qual inicialment no inclou paràmetres de QoS, els canvis produïts en la informació d'encaminament són tan sols deguts a modificacions en la topologia i connectivitat de la xarxa. En aquest entorn, donat que la freqüència en la qual s'espera rebre missatges advertint d'aquestes modificacions no és elevada, la majoria dels mecanismes d'actualització es basen en la inclusió d'un cert període de refresc. Així, les bases de dades s'actualitzen periòdicament mitjançant la distribució d'uns missatges que informen a la resta de nodes de l'estat de la xarxa, a fi de que cada node pugui actualitzar la seva base de dades.

No obstant això, hem de tenir en compte que en aquelles xarxes IP/MPLS altament dinàmiques amb requeriments de QoS, aquest mecanisme d'actualització basat en un refresc periòdic no serà útil. Això és degut a la rigidesa que presenta aquest mecanisme, la qual fa que no sigui aplicable a un entorn que presenti contínues variacions del paràmetres dels enllaços cada cop que s'estableixi o s'alliberi una connexió (ara a més de la topologia i connectivitat, s'inclouen paràmetres de QoS, com ampla de banda, retard, variació del retard, etc.). Per tot això, s'haurà de generar un mecanisme d'actualització molt més eficient que sigui capaç de mantenir les bases de dades dels nodes perfectament actualitzades reflectint els continus canvis en l'estat de la xarxa. L'alta granularitat d'aquest mecanisme provocarà una sobrecàrrega de la xarxa, degut a l'enorme quantitat de missatges d'actualització que seran necessaris per poder mantenir informació actualitzada en les bases de dades d'estat de l'enllaç en cada node.

Per reduir aquesta sobrecàrrega de senyalització apareixen les polítiques d'activació (triggering policies) que tenen per objectiu determinar en quin moment un node ha d'enviar un missatge d'actualització a la resta de nodes de la xarxa advertint-los de les variacions produïdes en els seus enllaços. Desafortunadament, l'ús d'aquestes polítiques d'activació produeix un efecte negatiu sobre el funcionament global de la xarxa. En efecte, si l'actualització de la informació de l'estat de l'enllaç en els nodes no es fa cada cop que aquesta informació es veu

modificada, sinó que es fa d'acord a una certa política d'activació, no es podrà garantir que aquesta informació representi de forma acurada l'esta actual de la xarxa en tot moment. Això pot provocar una selecció no òptima de la ruta seleccionada i un increment en la probabilitat de bloqueig de noves connexions a la xarxa.

Aquesta Tesi es centra en definir i solucionar el problema de la selecció de rutes sota informació inexacta o no acurada de la xarxa, problema conegut com "routing inaccuracy problem". Es consideren dos escenaris de treball, les actuals xarxes IP/MPLS i les futures xarxes òptiques basades en Wavelength Division Multiplexing (WDM). Per ambdós escenaris es proposa un nou mecanisme d'encaminament: BYPASS Based Routing (BBR) per xarxes IP/MPLS i BYPASS Based Optical Routing (BBOR) per xarxes WDM. Els dos mecanismes comparteixen un concepte comú, denominat "bypass dinàmic".

El concepte de "bypass dinàmic", permet que un node intermedi de la xarxa encamini el missatge d'establiment de la ruta que ha rebut del node origen a traves d'una ruta diferent a la que havia estat inicialment calculada per el node font (i explícitament indicada en el missatge d'establiment) quan aquest node intermedi detecti que inesperadament, l'enllaç de sortida no disposa de recursos suficients per fer front a les garanties de QoS requerides per la connexió a establir. Aquestes rutes alternatives, denominades "bypass-paths", són calculades per certs nodes de la ruta principal, simultàniament amb la ruta principal en el node generador del tràfic o d'entrada a la xarxa..

En xarxes IP/MPLS el mecanisme BBR aplica el concepte de "bypass dinàmic" a les peticions de connexions amb restriccions d'ampla de banda, mentre que en xarxes WDM el mecanisme BBOR l'aplica a l'hora d'assignar una longitud d'ona per la qual es transmetrà el tràfic.

Els dos mecanismes i els algorismes d'encaminament que se'n desprenen són avaluats i comparats en diferents escenaris de simulació, per verificar que redueixen de forma més que eficient els efectes negatius produïts sobre el funcionament global de la xarxa, com són la probabilitat de bloqueig i la selecció de rutes incorrectes, degut al fet de realitzar la selecció de rutes sota informació d'encaminament no suficientment actualitzada.

Table of Contents

List of Figures

List of Tables

Abbreviations

ASON	Automatic Switched Optical Networks
ATM	Asynchronous Transfer Mode
ATMF	ATM Forum
ALG	Algorithm
BA	Behaviour Aggregate
BBR	BYPASS Based Routing
BBOR	BYPASS Based Optical Routing
BDP	BYPASS Discovery Process
BGP	Border Gateway Protocol
BPA	Bypass-Path Address
BOSP	Balance-Obstruct-Sensitive Path Algorithm
BRP	Backward Reservation Protocol
CoS	Class of Service
CR	Constraint Based Routing
D-RSP	Dynamic-Restricted Shortest Path
DiffServ	Differentiated Services
DSCP	Differentiated Services Code Point
ERO	Explicit Routing Object
FRP	Forward reservation Protocol
GMPLS	Generalized Multiprotocol Label Switching
IE	Crankback Information Element
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
IS-IS	Intermediate System-Intermediate System
ISP	Internet Services Provider
LCP	Least-Congested Path Algorithm
LDP	Label Distribution Protocol
LSP	Label Switched Path
LSR	Label Switching Router
MCP	Multi-constrained Path Problem
MDWCRA	Maximum Delay-Weighted Capacity Routing Algorithm
MIRA	Minimum Interference Routing Algorithm
MP-BCP	Most-Probable Bandwidth Constrained Path
MP-DCP	Most-Probable Delay Constrained Path
MPLS	Multiprotocol Label Switching
MRP	Most Reliable Path
n_bp	Computed *bypass-paths* per route
NTDB	Network Topology Database
OP	Optimal Partition
OP-MP	Optimally Partitioned Most Probable Path
OSL	Obstruct-Sensitive Link
OSPF	Open Shortest Path First
OSSP	Obstruct-Sensitive-Shortest Path Algorithm
OSW	Obstruct-Sensitive Wavelength
OTN	Optical Transport Network
OXC	Optical Cross-Connect

PBR	Profile Based Routing
PHB	Per-Hop-Behaviour
PNNI	Private Network-to-Network Interface
PSR	Proportional Sticky Routing
QoS	Quality of Service
RFC	Request for Comments
RIP	Routing Information Protocol
RSP	Restricted Shortest Path
Rspec	Reservation Specification
RSVP	Resource Reservation Protocol
RTC	Routing Table Cache
RWA	Routing and Wavelength Assignment
SBR	Safety Based Routing
SLA	Service Layer Agreement
SOSP	Shortest-Obstruct-Sensitive Path Algorithm
SP	Shortest Path Algorithm
SSP	Shortest Safest Path Algorithm
SWP	Shortest-Widest Path
TBP	Ticket Based Probing
TE	Traffic Engineering
TED	Traffic Engineering Database
ToS	Type of Service
Tspec	Traffic Specification
VCI	Virtual Channel Identifier
VCR	Virtual Capacity Routing
VPI	Virtual Path Identifier
VPN	Virtual Private Network
WDM	Wavelength Division Multiplexing
WI	Wavelength-Interchangeable Network
WS	Wavelength-Selective Network
WSOSP	Widest-Shortest-Obstruct-Sensitive Path Algorithm
WSP	Widest-Shortest Path

Abstract

Traditional IP networks are based on the best effort model to transport traffic flows between network clients. Since this model cannot properly support the requirements demanded by several emerging real time applications (such as video on demand, multimedia conferences or virtual reality), some modifications in the network structure, mainly oriented to optimise network performance, are required in order to provide Quality of Service (QoS) guarantees.

Traffic Engineering is an excellent framework to achieve these network enhancements. There are two main aspects in this context that strongly interact with network performance: switching mechanisms and routing mechanisms. On one hand, a quick switching mechanism is required to reduce the processing time in the intermediate nodes. In IP networks this behaviour is obtained by introducing Multiprotocol Label Switching (MPLS). On the other hand, a powerful routing mechanism that includes QoS attributes when selecting routes (QoS Routing) is also required.

Focusing on the latter aspect, most QoS routing algorithms select paths based on the information contained in the network state databases stored in the network nodes. Because of this, routing mechanisms must include an updating mechanism to guarantee that the network state information perfectly represents the current network state. Since network state changes (topology) are not produced very often, in conventional IP networks without QoS capabilities, most updating mechanisms are based on a periodic refresh.

In contrast, in highly dynamic large IP/MPLS networks with QoS capabilities a finer updating mechanism is needed. This updating mechanism generates an important and non-desirable signalling overhead if maintaining accurate network state information is pursued. To reduce the signalling overhead, triggering policies are used. The main function of a triggering policy is to determine when a network node must advertise changes in its directly connected links to other network nodes.

As a consequence of reduced signalling, the information in the network state databases might not represent an accurate picture of the actual network state. Hence, path selection may be done according to inaccurate routing information, which could cause both non-optimal path selection and an increase in connection blocking frequency.

This Thesis deals with this routing inaccuracy problem, introducing new mechanisms to reduce the effects on global network performance when selecting explicit paths under inaccurate routing information. Two network scenarios are considered, namely current IP/MPLS networks and future WDM networks, and one routing mechanism per scenario is suggested: BYPASS Based Routing (BBR) for IP/MPLS and BYPASS Based Optical Routing (BBOR) for WDM networks. Both mechanisms are based on a common concept, which is defined as dynamic bypass.

According to the dynamic bypass concept, whenever an intermediate node along the selected path (unexpectedly) does not have enough resources to cope with the incoming MPLS/optical-path demand requirements, it has the capability to reroute the set-up message through alternative pre-computed paths (bypass-paths). Therefore, in IP/MPLS networks the BBR mechanism applies the dynamic bypass concept to the incoming LSP demands under bandwidth constraints, and in WDM networks the BBOR mechanism applies the dynamic bypass concept when selecting light-paths (i.e., selecting the proper wavelength in both wavelength selective and wavelength interchangeable networks).

Finally, the applicability of the proposed BBR and the BBOR mechanisms is validated by simulation and compared with existing methods on different network scenarios. These network scenarios have been selected so that obtained results may be extrapolated to a realistic network.

PART I

INTRODUCTION

In this introductory Part, the motivation, goals and structure of the Thesis are presented. The current Internet scenario including concepts related to Quality of Service, such as routing mechanisms, new network architectures and signalling protocols are also introduced in this Part so that the problem to be addressed in next Part can be easily allocated in the current network scenario.

Chapter 1

Objectives and Structure of the Thesis

The network conception has extraordinarily evolved from that day in September 1969, not so far in the short network history, when Leonard Kleinrock went down so well at the challenge of sending a message from his Host Computer at UCLA. From that day when the first message containing the word “LOGIN” partially succeed on reaching the destination node (indeed destination node only got the word “LOG”) to the current Internet, network components, network connectivity, network applications and network utilization have constantly evolved in a way that maybe not even more optimistic might imagine.

Currently, the Internet does not provide *Quality of Service* (QoS) guarantees, and data delivery is based on a simple best-effort transmission model. Emerging real-time applications, such as video on demand, multimedia conferences or virtual reality, cannot be supported under this network definition, due to both the variable delays in the queuing process and the problem of congestion. Before these

applications can be used, the network has to be enhanced to support end-to-end QoS. This enhancement lies in optimising network performance to provide QoS guarantees by improving the utilization of network resources (*Traffic Engineering*, *TE*) [1], and providing resilience features for quick recovery from failures. In the *TE* context there are two determinant factors: to have a quick switching mechanism and to have powerful routing mechanisms to select the path by which traffic is sent. The first objective can be reached by implementing *Multiprotocol Label Switching* (*MPLS*) [2] and the second can be achieved by adding new routing mechanisms oriented to improving network performance.

MPLS is an advanced, connection-oriented forwarding scheme, which allows streams from any particular ingress node *(Label Switching Router*, *LSR*, in an *MPLS* domain) to any particular egress node to be individually identified with a simple label. Therefore, *MPLS* provides a straightforward mechanism to forward the traffic associated with each ingress node to egress node pair and substantially improves source routing, since the IP address of the intermediate nodes need not be piggybacked on each packet along the end-to-end path (*Label Switched Path*, *LSP*, in an *MPLS* domain).

As a basic definition, it is possible to say that the main routing objective is to drive packets to the right destination. In order to perform this goal, the routing process selects the path that can best transport the traffic from the source node to the destination node. This selection is done in accordance with the network state information, namely the current network topology in traditional IP routing, which is mostly obtained from several databases where this information is maintained. In traditional hop-by-hop routing each intermediate node instantly decides where the current packet should be sent in order to reach the destination according to the database information existing in the node. However, if routing is explicitly done in the source nodes (i.e., source routing), only the source node's routing information is considered. It should be noticed that source routing leads to a reduction in both network control complexity and in the number of databases checked when selecting a path.

Traditional IP routing algorithms are OSPF [3], RIP [4], IS-IS [5] and BGP [6]. Basically, they are based on computing the shortest path by applying either Dijkstra's algorithm or the Bellman-Ford algorithm. The first is a link state algorithm and the second a distance-vector algorithm. The databases mentioned above are filled with link state information or distance information, respectively. In any case, both algorithms ensure only a best-effort performance, which is not recommended for applications demanding specific QoS guarantees. In fact, two network models might be described: on one hand, a network that supports the best effort transmission model for traffic without QoS requirements; and on the other hand, a QoS network model necessary to support traffic with QoS constraints.

Currently, there are several QoS routing algorithms proposed in the literature to cope with the QoS network model that improves the best-effort transmission model currently used in Internet. Unlike traditional IP routing algorithms, for QoS provisioning the routing algorithm must take into account more parameters than exclusively those related with topology and connectivity. QoS routing algorithms include QoS parameters in the path decision process to select the most suitable path in accordance with both traffic requirements and network state, including network topology (quasi-static) and the resources available at each node (dynamic). The management of these parameters is done by the routing protocol, which must include a mechanism to collect, distribute and update all of the parameters needed by the QoS routing algorithm.

1.1 Thesis Motivation

Important factors in the global routing behaviour are where and how routing decisions are taken. Hence, assuming source routing, if resource availability is a key factor in the path selection process, the QoS parameters used by the routing algorithm to decide the routes must perfectly represent the current network state. Two different aspects must be considered to guarantee accurate QoS routing.

First, a mechanism to keep the network state information perfectly updated must be included in the routing protocol. The main function of this mechanism is to decide when a node must send update messages throughout the network to advertise state changes in its directly connected links to all the other nodes. In traditional IP

networks, without QoS requirements, network state changes are only due to topology variations, which are not often expected. Hence, maintaining accurate information in the network state databases is easily achieved without many update messages. However, in *IP/MPLS* networks with QoS requirements, keeping the link state databases perfectly updated involves sending update messages whenever a new *LSP* is established or an existing one is released.

Therefore, in a large connection-oriented packet-switched network scenario, where changes are produced very often, this updating process generates a non-desirable signalling overhead. To reduce such a signalling overhead, one of several currently available triggering policies is applied. An unfortunate consequence of applying any of these triggering policies is that the information contained in the network state databases might not represent the current network state at the path set up time. This is only one of the possible causes of inaccurate routing information. When routing is done under inaccurate routing information, the route selected according to this information can be blocked in the path set up process (*routing inaccuracy problem*). This is due to the possibility that the resources required by the incoming *LSP* are not available, contrary to what the database on the source node states.

Second, in addition to the problem above, most QoS routing algorithms utilize the nominal available bandwidth information of the links to select paths. This routing information is obtained from the link state databases stored on each node, which are updated regarding traffic requirements included in new *LSP* demands by applying any triggering policy. However, assuming that most of the clients generating network traffic do not completely use the requested bandwidth, that is, do not strictly fulfil the *Service Layer Agreement* (SLA), a difference exists between the nominal link utilization and the actual link utilization. This gap leads to non-efficient network resource utilization, since the path selection process is performed according to nominal link state information instead of actual link utilization. This problem is addressed in [7], where the authors propose a path selection scheme based on obtaining more accurate link utilization information. Initially, this link utilization information may be obtained from the link state databases. However, this is not possible over time due to scalability concerns. In fact, in this scenario an update

message must be sent not only when an *LSP* is established or released, but also when the actual bandwidth used by the traffic flowing by an already established *LSP* changes.

This degree of granularity cannot be supported by any network, so a different method to collect the actual link utilization information is required. The authors propose a method based on estimating the future link utilization. They indeed suggest an algorithm, the *Available Bandwidth Estimation Algorithm,* for computing both an estimation of the real available bandwidth on each link and the duration for which the estimate is valid. This prediction is obtained by sampling the network state periodically, with the period of time changing dynamically depending on the traffic characteristics and the required conservatism of the network domain.

Based on the estimated values obtained by applying this algorithm, a new path selection algorithm is suggested that uses the estimated available bandwidth values as weights of the links, which then are used by a shortest widest path routing algorithm to select the optimal path. Finally, in order to limit network congestion, a threshold parameter is added. Once the path has been computed by using the modified shortest widest path, the routing algorithm computes the available bandwidth on the bottleneck link of the path. Then the threshold parameter is applied to this bottleneck value to compute a benchmark for path selection in such a way that if the bandwidth requested is larger than a certain fraction of the bottleneck link bandwidth, the incoming request is rejected. The authors show that the proposed path selection algorithm performs better than the shortest widest path routing algorithm, because the proposed routing algorithm based on the Available Bandwidth Estimation Algorithm has more accurate information about the actual link load.

1.2 Thesis Objective

Two different aspects have been discussed related to the routing information used to select the routes so far. Both aspects may be summarized as the effects produced in global network performance when using either inaccurate link state information or inaccurate link utilization information. This Thesis focuses on addressing the first scenario, namely the routing inaccuracy problem produced when the path selection process is performed under inaccurate nominal link state information. From this

point, the expressions nominal link state information and link state information are used with the same meaning in this Thesis. Hence, explicit QoS routing is considered, where the explicit paths are computed, selected and established by the source node (or ingress node), assuming that the path decision is taken based on inaccurate nominal link state information.

IP/MPLS networks are being considered so far. However, it is well known that the network model is evolving to an *Optical Transport Network* (*OTN*). Optical Transport Networks based on *Wavelength Division Multiplexing* (*WDM*) appear as a potential solution to cope with the increasingly growth of Internet traffic demands. In such systems all-optical *WDM* channels are used to allow the end-to-end users communication. These *WDM* channels are referred as lightpaths, and must be selected in a proper manner in order to optimize the network resources. It is in this point where the routing becomes an important factor in the global network performance. Once more, the accuracy of the network state information is a key aspect to be considered when selecting lightpaths.

The main goal of this Thesis is to propose a new routing mechanism able to reduce the impact on global network performance because of selecting routes under inaccurate routing information. Two mechanisms, *BYPASS Based Routing* (*BBR*) and *BYPASS Based Optical Routing (BBOR)* are proposed in this Thesis to address the routing inaccuracy problem in *IP/MPLS* and *WDM* networks respectively. The *BBR* mechanism contributes to the improvement of global network performance in terms of a substantial connection blocking reduction and a more optimal path selection when the incoming traffic requires bandwidth guarantees. Four routing algorithms inferred from the *BBR* mechanism are evaluated by simulation. The *BBOR* mechanism modifies route selection and wavelength assignment in such a way that connection blocking is reduced as well. Again, three routing algorithms inferred from the *BBOR* mechanism are also evaluated by simulation.

1.3 Thesis Structure

This Thesis is organized in four Parts each one divided into several Chapters. A brief description of each one is now presented.

PART I: INTRODUCTION

Chapter 2: After introducing the network scenario in Chapter 1, this Chapter introduces the main concepts, goals and capabilities of *Traffic Engineering* (*TE*). Main features of proposed QoS network architectures, such as *Integrated Services* and *Differentiated Services* are also briefly introduced. Then, skills and drawbacks of the *Resource Reservation Protocol* (*RSVP*) are also presented. Finally, the use of *Multiprotocol Label Switching (MPLS)* is justified based on the benefits on global network performance due to its application. The main goal of this Chapter is to define the working scenario, focusing on the network modifications in terms of structure and architecture needed to cope with current network challenges.

Chapter 3: A main aspect in the network evolution concerns to QoS Routing. Chapter 3 finely focuses on this topic. Routing evolution, from traditional IP routing to QoS routing, and main routing concepts are described in this Chapter. Most recent routing algorithm proposals existing in the literature are shown to illustrate routing evolution and main routing problems.

PART II: QoS ROUTING IN IP/MPLS NETWORKS

Chapter 4: In Chapter 4, the problem addressed in this Thesis, called *routing inaccuracy problem*, is introduced. Causes and origins that motivate the existence of this problem are clearly presented and justified in this Chapter.

Chapter 5: Some solutions exist in the literature addressing the *routing inaccuracy problem*. A clear and extensive description of these solutions is presented in Chapter 5.

Chapter 6: Then, Chapter 6 presents a routing mechanism called *BYPASS Based Routing (BBR)* as the solution proposed in this Thesis to cope with the *routing inaccuracy problem*. Two new routing algorithms are inferred from the *BBR* mechanism in this *IP/MPLS* scenario. Different simulations are evaluated to verify the benefits of these routing algorithms.

Chapter 7: After that, in Chapter 7, the *BBR* mechanism is applied under bandwidth constraints, generating two new routing algorithms which are also evaluated by simulation and compared with previous *BBR* algorithms.

Chapter 8: The *BYPASS Discovery Process* (*BDP*) is proposed in Chapter 8 to improve the *BBR* performance. The *BDP* extends the *BBR* applicability, so improving the obtained benefits.

PART III: ROUTING IN WDM NETWORKS

Chapter 9: This Chapter shows the expected evolution in network technology, network usage and network requirements that ends in the introduction of optical networks. Main concepts in optical networks, such as those related to routing and the definition of a Control Plane are also described.

Chapter 10: This Chapter serves to introduce the problem. Once the advantages of applying the *BBR* mechanism in an *IP/MPLS* scenario have been analyzed, the problem is extended to optical networks. Finally, work existing in the literature copying with the *routing inaccuracy problem* in optical networks is discussed.

Chapter 11: The *BYPASS Based optical Routing (BBOR)* is proposed as a solution to address the *routing inaccuracy problem* in optical networks. The *BBOR* is based on extending the main concepts of the *BBR* mechanism. Unlike the *BBR* mechanism the *BBOR* also proposes a new triggering policy to reduce the signalling overhead. This Chapter focuses on applying the *BBOR* to wavelength selective networks where wavelength conversion is not permitted.

Chapter 12: After evaluating the *BBOR* behaviour in wavelength selective networks, the *BBOR* is modified to be applied on wavelength convertible networks. Again, the benefits of applying the *BBOR* are verified by simulation.

PART IV: CONCLUSIONS AND FUTURE WORKS

Chapter 13: In this Chapter, the main contributions of the Thesis are pointed out. A short summary is also introduced to conclude the Thesis.

Chapter 14: Finally, the future work is presented. In this Chapter, future lines of work, some of them already planned are briefly described. It is worth to notice that because of the nature of the Thesis which proposes two routing mechanisms to be applied on two different network scenarios, future research is also suggested in both network scenarios and to interoperate between both scenarios.

This Thesis ends with two Appendix. Appendix A lists publications and projects related to this work. Appendix B briefly describes modifications needed on the network simulator to provide it with *BBR* and *BBOR* capabilities

Chapter 2

QoS and TE in the new Internet

Implementing *Traffic Engineering* (*TE*) is fundamental in the current network models, mainly because current routing protocols forward traffic through the shortest paths. *TE* capabilities can be provided by the *Multiprotocol Label Switching* (*MPLS*). In fact, as explained later on the main, direct and most significant benefit provided by the *MPLS* is the provision of *TE* capabilities. The main target of *TE* is to control how traffic flows through one's network so as to optimise network performance and resource utilization [1]. Traditional shortest path selection leads to congestion on some links along the selected path while longer paths are under-utilized. Congestion might be reduced by acting on the routing metric, but this solution may only be suggested on a small network scenario. In a large *Internet Service Provider* (*ISP*) network, new tools are needed for *TE* provisioning. *TE* provides the network with three main aspects: includes a guaranteed QoS, improves the utilization of network resources and provides for quick recovery when a node or link fails. Traditional IP routing mechanisms based on the shortest path selection are extended to balance, distribute and optimise the networks resources turning out QoS routing. Moreover,

developing fast rerouting mechanisms is vital to address network components failures. New overall QoS architectures, such as *Integrated Services* (*IntServ*) and *Differentiated Services* (*DiffServ*) are needed to support a QoS network scenario. Even though this Thesis focuses on the first aspect a short description of both architectures is now presented just to introduce the current networking scenario.

2.1 Integrated Services model

The *IntServ* model, developed by the *Internet Engineering Task Force* (IETF), provides an end-to-end QoS solution. *IntServ* defines a set of service classes which specify the potential needs of different clients. *IntServ* follows the signalled QoS model, in which network resources are reserved according to the QoS needs signalled by the end clients. Therefore, end-to-end QoS is obtained by way of end-to-end signalling, state maintenance for each signalled session and admission control at each network element. Two parameters are mainly defined, a traffic specification called *Tspec* and a reservation specification called *Rspec*. The former specifies the kind of application traffic used by the client and so, incoming to the network. The latter specifies the required level of QoS and the reservation of network resources. In accordance with these specifications, *IntServ* requires network elements such as routers, to perform the following:

- policing functions to verify that incoming traffic is conformed to its *Tspec*. Packets that do not meet the *Tspe*c values are dropped.
- admission control functions to check if there are enough resources to support the QoS traffic requirements. If available resources are not enough, the incoming request is rejected.
- packet classification functions, queuing and scheduling mechanisms to separate and properly handle those packets demanding for a specific level of QoS.

There are two service classes defined in the *IntServ*, guaranteed service and controlled load. The guaranteed service class provides for strict bounds on end-to-end delay and assured bandwidth for traffic that meet the requested QoS

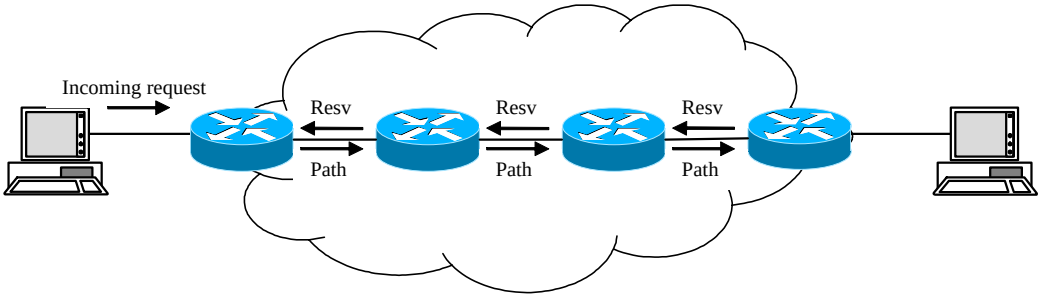


Figure 1. *Path* and *Resv* messages

specifications. The controlled load service class provides a better-than-best effort and low delay service under light to moderate network loads.

A signalling protocol must also be specified in the *IntServ* network model to perform its functions. The *Resource Reservation Protocol (RSVP)* [8] is an *IntServ* signalling protocol used by both the end clients to demand their QoS needs according to the defined *IntServ* service classes and the core network to handle the path establishment. The *RSVP* must be implemented in all the network elements, to allow clients to demand their specific QoS levels. *RSVP* was born to define, establish and maintain reservation of those resources required by a certain *LSP* incoming demand to succeed on flowing traffic. There are two main messages carrying the *RSVP* information: *Path* and *Resv*. Figure 1 illustrates both messages. Once the source node, applying a certain routing algorithm, computes a route reacting to an incoming *LSP* demand, it sends a *Path* message which is forwarded downstream across the selected path to reach the destination. *Path* message includes an *Explicit Routing Object* (*ERO*) including the IP address of the intermediate nodes across the selected path, and *Tspec* and classification information provided by the source node. When the *Path* message reaches the destination node, this node sends a *Resv* message back to the sender along the reverse path, identifying the session for which the reservation is to be made. The reservation process ends with a new path where the traffic associated with the incoming *LSP* demand flows or with a rejection message when there are not enough available resources to cope with the QoS traffic requirements.

However, the *IntServ* and *RSVP* applicability was not really extended due to scalability concerns. In fact, a soft-state must be maintained on all nodes along the selected route to keep the resource reservation alive. This implies a large number of

signalling messages flowing throughout the network. Hence, *IntServ* may only be applied to small networks (Intradomain Routing). Even though *RSVP* is really not used to perform resource reservation, it may be used as a signalling protocol.

2.2 Differentiated Services model

As the Internet traffic and the diversity of applications grow, different QoS levels must be applied to different traffic flows demanding for specific differentiated services. The *Diffserv* model [9] for IP QoS provisioning has also been proposed by the *IETF* to allow the network to support different QoS levels according to the QoS required by the end user. This model is very similar to the IP precedence model. The IP precedence model handles traffic by classifying various traffic flows into aggregated classes. The appropriate QoS is required for each aggregate class. Three bits in the *Type of Service* (*ToS*) field stand for eight different aggregated classes. As in the IP precedence model, *DiffServ* model divides traffic into a small number of classes and allocates resources on a per-class basis. This classification is made based on the information contained in the *Differentiated Services Code Point* (*DSCP*) byte. The *DSCP* is carried in the *ToS* or in the *Class of Service* (*CoS*) fields in the IPv4 or IPv6 header respectively. According to [10] only 6 weighted bits are meaningful in the *DSCP* byte while the last 2 bits are currently not used. Hence, even though 64 different classes might be implemented, in practice only a few classes are really implemented. The above defined eight IP precedence levels can be mapped to a fixed *DSCP* classes as shown in Table 1.

Table 1. IP precedence values mapped to *DSCP*

IP Precedence	*Name*	*DSCP*
0	Routine	DSCP 0
1	Priority	DSCP 8
2	Immediate	DSCP 16
3	Flash	DSCP 24
4	Flash override	DSCP 32
5	Critical	DSCP 40
6	Internet control	DSCP 48
7	Network control	DSCP 56

Packets are marked to be properly handled across the path. When the ingress node receives an IP packet it sets the *DSCP* to identify this packet to the desired service

class. Then, intermediate nodes across the path check the *DSCP* field value and determine the QoS required by this packet, as shown in Figure 2. This is known as *Per-Hop-Behaviour* (*PHB*). The *PHB* includes all the mechanisms (i.e., packet scheduling, queuing, policing or sapping behaviour) applied in a node to provide the packet for the required QoS. *Behaviour Aggregate* (*BA*) is defined as the set of packets which traversing the same node have the same *DSCP*.

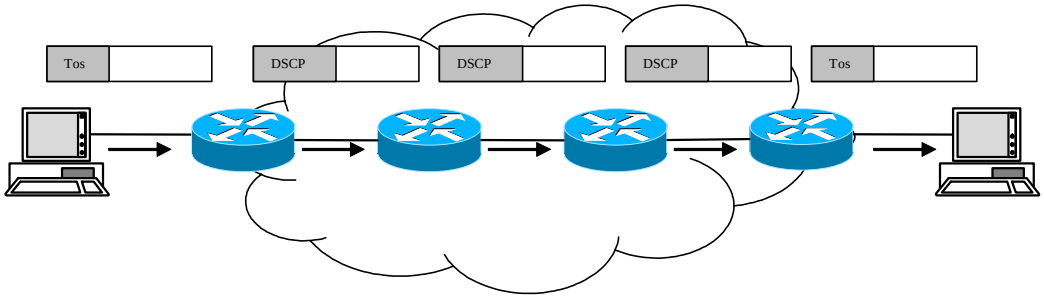


Figure 2. *PHB* in the *DiffServ* model

Four *PHB* implementations are currently available:

- Default *PHB*: In this case only best effort delivery is guaranteed. Packets marked with a *DSCP* value that cannot be mapped to a *PHB* are directly mapped to this *PHB*.
- Class-selector *PHB*: This *PHB* allows IP precedence model to be compatible with the *DiffServ* model.
- Expedited Forwarding (*EF*) *PHB*: Packets marked with the *EF* are prioritized for delivery over others, by providing for low packet loss, low latency, low jitter and guaranteed bandwidth service.
- Assured Forwarding (*AF*) *PHB*: This *PHB* specifies an *AF* class and drop precedence for IP packets, in such a way that different forwarding assurances are given. There are four classes, each one specifying three drop precedence values. In case of congestion packets are dropped based on their relative drop precedence values within the *AF* class.

There are several main characteristics which ease *DiffServ* implementation:

- IP must not be modified as packets are marked in the edge node by using either the *ToS* or *CoS* fields in the IPv4 or IPv6 header respectively.
- Network applications must not be modified to implement *DiffServ* model as an *SLA* has been previously agreed between network provider and network client.
- It is easily scalable due to the aggregation mechanism. In fact traffic matching the same *DSCP* value is handled in the network as an aggregate.
- Intermediate nodes must not maintain a soft-state including information about packet flows as packets are individually handled at intermediate nodes.

2.3 Multiprotocol Label Switching

IntServ and *DiffServ* architectures can be implemented using the *Multiprotocol Label Switching* (*MPLS*). *MPLS* is an advanced label-based switching mechanism which uses the information contained in the labels to compute the routes. In the first case, *IntServ* model, bonds are created between labels and *RSVP* flows, to identify the type of resource reservation associated with the traffic. In the second case, *DiffServ* model, the appropriate *PHB* must be determined from the label. A field is included in the *MPLS* shim header to allow *MPLS* to support eight different *DiffServ* classes.

In an *IP/MPLS* scenario, packets are classified and routed at the ingress *LSR*s. Then, an *MPLS* header is inserted. Depending on the technology, frame-based or cell-based, 32-bit labels are embedded in this header between the Layer 3 header and the Layer 2 header or in the *Virtual Path Identifier* (*VPI*) and *Virtual Channel Identifier* (*VCI*) fields, respectively. When an *LSR* receives a packet it uses the label as the index to look up the forwarding table and the packet is forwarded according to the forwarding table entry. Then the incoming label is replaced by the outgoing label and the packet is forwarded to the next *LSR*. It is worth to notice that the label allocation and distribution is made at the path set-up time.

MPLS uses a label distribution mechanism, the *Label Distribution Protocol* (*LDP*) that handles *LSP*s set-up and any other negotiation between *LSR*s. Currently, a number of different label distribution protocols are being proposed. Existing protocols have been extended so that label distribution can be piggybacked on them

(e.g. in MPLS-RSVP [11] the *RSVP* is extended with several additional objects to allow the establishment of explicitly routed *LSP*s using *RSVP* as a signalling protocol). Moreover, new protocols have also been defined for the explicit purpose of distributing labels, such as MPLS-LDP [12] and MPLS-CR-LDP [13].

Although initially the main goal of label-based switching mechanisms were to export the speed of Layer 2 to Layer 3, it is really not the main benefit of such mechanisms because of the low time required by newer Layer 3 switches to perform routing. However, as routing decisions are taken according to labels instead of IP addresses, the main benefits introduced to the IP networks by the *MPLS* are:

- Virtual Private Networks (*VPN*): *MPLS* allows providers to create Layer 3 *VPN*s through their own backbone network for multiple customers, with no encryption required.
- Traffic Engineering: *MPLS* provides traffic engineering capabilities needed for the efficient use of network resources. This feature optimises global bandwidth utilization.
- Quality of Service: *MPLS* allows service providers to provide multiple classes of service with hard QoS guarantees.
- Integration of IP and ATM: *MPLS* allows those carriers networks employing an overlay model in which ATM is used at Layer 2 and IP at Layer 3, to migrate many of the functions of the ATM Control Plane to Layer 3, so reducing scalability issues and simplifying network provisioning.

It is worth to note that *MPLS* includes the main advantages of Layer 3 and layer 2, i.e., performance and scalability respectively. The huge and almost incredible evolution on both, network utilization and network applications lead to a continuous endeavour to keep the network infrastructure up-to-date. *MPLS* allows service providers to differentiate services in a QoS scenario, without requiring continuous modifications in the existing network infrastructure.

MPLS Traffic Engineering uses the *RSVP* to automatic establish and maintain a tunnel, *LSP* tunnel across the backbone. In this scenario the *RSVP* is used only to signal the path set-up. The path of the *LSP* tunnels is selected based on the incoming

traffic requirements and the available network resources. The selection process is performed at the source router, by using *Constraint-based Routing* (*CR*). Unlike traditional *IP/MPLS* routing where routes are selected according to network topology, *CR* selects routes considering multiple constraints, such as *LSP* and link attributes. As a consequence, network load is distributed more fairly. From the network client point of view, this traffic traverses the *MPLS* backbone through an end-to-end tunnel which connects the source and the destination nodes. Paths are properly selected whenever network information used by the *CR* accurately represents the real network state in terms of QoS parameters. This constraint-based information is disseminated across the *MPLS* network by extending existing link-state routing protocols such as *OSPF* or *IS-IS* to generate *Traffic Engineering Databases* (*TEDs*). Routing protocols based on link state network information perform better than those based on distance vector network information because unlike the first ones the second ones do not include enough information in their routing tables to compute alternative paths needed by *TE*. Therefore, *OSPF* and *IS-IS* are extended to properly carry constraint-based information.

As a summary, *TE* is essential for service provider and Internet service provider backbones because both backbones must support a high capacity traffic demands and a quick response to network failures. *MPLS* is very useful in the *TE* scenario since it allows service providers to offer traffic engineering networks without substantial network modifications. The main advantages of *MPLS Traffic Engineering* are the following:

- When using *MPLS*, the Layer 3 integrates traffic engineering capabilities, so optimising the path selection process, according to the available bandwidth capacity and the network topology.
- The routing decisions are taken based on the available network resources and the resources required by the incoming *LSP* demand.
- *MPLS Traffic Engineering* uses *Constraint-based Routing*. *CR* selects that path among the shortest ones that meets the QoS requirements.
- *MPLS Traffic Engineering* includes a mechanism to switch traffic flow over when a network element fails.

- *MPLS Traffic Engineering* enables unequal-cost load sharing.
- It computes explicit routes accounting for link bandwidth and for the size of the traffic flow.
- Explicit routing is optimised (in comparison with IP source routing) since a single label instead of the complete list of intermediate IP addresses is sent in the set-up message.
- Explicit paths are dynamically set up by an automated signalling process.

Chapter 3

QoS Routing

New multimedia applications are appearing over the Internet, demanding particular QoS requirements, such as bandwidth, delay, jitter, packet loss and reliability, which must be taken into account when selecting paths. As mentioned in the last Chapter different QoS architectures, such as *DiffServ* and *IntServ*, are proposed to meet these QoS requirements. A key aspect in these QoS architectures is the routing process, i.e., how routes are computed, selected and established.

It is worth to notice that there are two different entities when talking about routing, the routing protocol and the routing algorithm. The routing protocol attends to the matter of collecting the network state dynamics and to flood this information throughout the network. Based on this network state information, the routing algorithm selects the optimal path.

Traditional IP routing algorithms, which are based on the best-effort transmission model, select routes according to the shortest path routing. These algorithms select the path that optimizes the sum of a single value, such as hopcount or delay along the

selected path. This routing model is not suitable in a QoS environment. When a certain guarantee is required for sending a particular traffic flow, routing algorithms must add some QoS attributes to the path selection process. Unlike shortest path based routing algorithms, QoS routing algorithms select that route which more precisely meets multiple QoS requirements. Basically, the main goal of QoS routing is to find a route for a particular traffic flow with certain QoS requirements conforming to the QoS cost parameters, which specify the available resources in the network (i.e., conforming to the network resources that can be used to support the incoming traffic request). These QoS requirements may be bottleneck requirements, such as bandwidth, or additive requirements, such as end-to-end delay, in which case the QoS routing process looks for that path that guarantees a minimum available bandwidth or an end-to-end delay bound respectively. In order to perform this route evaluation and selection, the link state databases are extended to include information on available resources, and are often referred to as *Traffic Engineering Databases* (*TED*). In addition to QoS guarantees, there are some other widely sought solutions to common networking challenges that can be perfectly synthesized with the above mentioned goals: optimisation of network utilization, load distribution, the number of paths successfully routed, etc. There is currently a concerted effort in the networking community to achieve all of these objectives.

In the QoS routing context, there are two main issues to be addressed. Firstly, routing decisions are taken based on the network state information. Each node collects this information by implementing a flooding mechanism, which disseminates this information throughout the whole network. Despite its simplicity and reliability, flooding involves unnecessary communications and causes inefficient use of resources, particularly in highly dynamic network where frequent distribution of multiple QoS parameters is expected. Secondly, once nodes have updated network state information a routing algorithm is applied to select the optimal route. It is necessary to point out that a node contains updated network state information when this information perfectly represents the real network state at the moment when path selection process is performed.

Being aware that QoS routing is essential in a network architecture that needs to satisfy traffic and service requirements, it must be assumed that the process required to manage the path selection turns out a difficult problem to be solved.

Computing paths based on multiple QoS constraints is called *multi-constrained path selection problem* (*MCP*). In general *MCP*, is known as an NP-complete problem, therefore it is intractable for large networks. There are several algorithms proposed in the literature to address this problem. Most important MCP algorithms are *Jaffe's Approximate Algorithm* [14], *Iwata's Algorithm* [15], *Self-Adaptive Multiple Constraints Routing Algorithm (SAMCRA)* [16], *Chen's Approximate Algorithm* [17], *Randomized Algorithm* [18], *H_MCOP* [19], *Limited Path Heuristic* [20], and *A*Prune* [21]. A performance evaluation of these algorithms can be found in [22] where fundamental concepts involved in QoS routing are deduced based on the simulation results.

In addition to these algorithms there are other works in the literature aiming at addressing special important sub-problems in QoS routing, such as QoS routing in the context of bandwidth and delay, which is not NP-complete. This covers works such as the *Widest-Shortest Path* (*WSP*) [23], the *Shortest-Widest Path* (*SWP*) [24] and the *Maximum Delay-Weighted Capacity Routing Algorithm* (*MDWCRA*) [25]. In the *WSP* links with residual bandwidth lower than the requested bandwidth are pruned, therefore generating a reduced graph containing only those links supporting the incoming traffic demands. This reduced graph is used to select the shortest path. When there are multiple shortest paths available, the path that maximizes the minimum residual bandwidth on the links in the path is selected. The *SWP* algorithm performs similarly to the *WSP*. It selects the shortest path among the widest ones. In [25] authors concentrate on the specific problem of designing bandwidth-delay constrained algorithms taking into account knowledge of the source-destination node pairs. When QoS routing considers delay and cost is known as *Restricted Shortest Path* problem (*RSP*), which is NP-complete. Focusing on this issue are works presented in [26], [27], [28] and [29]. In [30] a selection of different path selection algorithms based on combining bandwidth, delay and cost (in terms of number of hops) can be found. Reference [31] considers pre-computation of paths with

minimum hopcount and bandwidth guarantees. The effects of reserving in advance of the path selection process are addressed in [32].

There are other significant contributions, focusing on other aspects to select routes. The *Maximally Disjoint Shortest and Widest Paths* [33], selects more than one feasible path. In [34] there are proposed bandwidth guaranteed dynamic routing algorithms. The *Minimum Interference Routing Algorithm* (*MIRA*) [35] selects optimal paths not only based on bandwidth guarantees but also considers ingress/egress characteristics in order to avoid negative interferences between routes sharing common links. Inspired by the *MIRA*, the *Profile-Based Routing* (*PBR*) [36] selects routes by using a "traffic profile" of the network as a rough predictor of the future traffic distribution.

QoS routing is generally blamed for increasing the complexity of the path selection process. Several factors can drive one to this conclusion. First, several new parameters used to provide the network with QoS capabilities are added to the routing process. These new parameters generate two implications, namely the database structure must be properly augmented to allocate them and the computational time will also be augmented since more parameters must be considered. Second, the number of update messages needed to maintain perfectly up-to-date link state databases could reach levels that negatively impact correct network performance. The first issue is not of excessive importance, since successive technological advances reduce the cost impact of QoS routing. The second issue has a fundamental influence on global network performance, since the number of update messages cannot be reduced without generating collateral negative effects that can hugely deteriorate network performance. As in IP routing, in QoS routing path selection is performed in accordance with the link state information (assuming Dijkstra's algorithm) contained in the databases of each node in the network. In order to optimise the path selection process the link state information must be correctly updated so that it accurately represents the current network state. In order to perform this, updating mechanisms are incorporated in the routing protocol. These mechanisms allow any node to send update messages that will be flooded throughout the network, advertising the latest changes in its directly connected links. In this way, global network state information is maintained in every node. The important

influence that these updating procedures have on global network performance and their impact on the number of paths successfully routed will be later shown.

In fact, most current QoS routing algorithms assume as a condition that the network state databases from which the routing tables are built represent a current picture of the network state. However, it is possible that due to some circumstances this information does not perfectly represent the real network state. In this case a certain degree of routing uncertainty or routing inaccuracy exists in the network state information. Moreover, if update messages are not enough fast flood routing instabilities may occur which produce undesirable routing oscillations. An initial approach to address this problem is based on advertising link weights that are properly quantified instead of instantaneous values [37] and [38]. However, in certain scenarios and under high loads or bursty traffic these mechanisms do not enough reduce routing oscillations. Algorithms for load balancing avoid routing oscillations by providing multiple paths from a source and a destination. Some works related to this can be found in [39], [40], [41], [42], [43], [44] and [45].

As a summary, this Part serves to introduce main aspects related to QoS, such as QoS network architectures and QoS routing algorithms. This part is very useful to describe the network scenario where this Thesis is placed.

PART II

QoS ROUTING IN IP/MPLS NETWORKS

This Part gives a clear analysis of what is the research problem tackeld in the Thesis as well as a broad overview of most recent and significant contributions existing in the literature focussing on the introduced problem. Along the different Chapters of this Part the proposed routing mechanism is formally stated, illustrated with different network examples, enhanced to include bandwidth constraints and finally evaluated in several network scenarios.

.

Chapter 4

The Routing Inaccuracy Problem in IP/MPLS Networks

Routing algorithms select routes based on the network state information contained in the network state databases, called *TED*s when including *TE* attributes. These *TED*s must be updated in order to include the network evolution, therefore modifying the network state information. Routing protocols must include an updating mechanism devoted to collect, distribute and maintain accurate network state information on each node along the network. However, maintaining accurate network state information cannot be always guaranteed, therefore generating inaccurate network state information. Separating two different components in the updating distribution process, namely the number of nodes and links able to generate update messages and the frequency at which these nodes and links generate these messages, leads to an initial classification of the origins of routing inaccuracy.

In a large, connection-oriented packet-switched network scenario it is extremely difficult to guarantee that the information contained in the network state database on

each node has been perfectly updated, due to the network dynamics and the huge amount of signalling messages needed to maintain accurate state information in all network elements. Some proposals exist, such as the *Private Network-to-Network Interface* (*PNNI*) [46] introduced by the ATM Forum to distribute network state information in a scalable form, based on a hierarchical process that allows the network to aggregate network state information based on the hierarchical level context.

However, this aggregation process implicitly introduces a loss of information, since information about physical nodes and links is not distributed. This loss of information introduces a certain degree of inaccuracy in the link state information, which cannot be easily measured since it depends on the aggregation method in use. For example, if an entire network is aggregated to a single virtual node, the available bandwidth of this virtual node can be obtained as either an average of the individual bandwidths or the best individual bandwidth or the worst individual bandwidth. In each case some information is lost, but the degree of inaccuracy is different. In fact, the larger the number of hierarchical levels the larger the inaccuracy.

Another important factor to be considered when analysing the causes of routing inaccuracy is the frequency at which update messages are sent throughout the network. In highly dynamic networks (plenty of *LSP*s are being constantly established/released) when link state changes are frequent, it is impractical to keep the link state databases correctly updated due to the large number of signalling messages needed to successfully perform this process. These messages generate an unnecessary reduction of available bandwidth in all links and consume processing cycles in all nodes. In order to reduce the frequency of these link state advertisements it is important to properly define when a node should trigger a message to inform the network about changes in one or more of its directly connected links. Hence, in order to reduce the number of update messages some triggering policies have to be defined; in this way routing accuracy varies depending on the values assigned to the triggering policies parameters. On the one hand, having the network perfectly updated leads to an increment in the number of update messages (signalling overhead), and on the other hand reducing the overhead leads to an increase in

routing inaccuracy. In fact, a trade-off exists between having accurate link state information and reducing signalling overhead.

Based on the mechanism used to trigger the update messages, the above-mentioned triggering policies can be classified as follows [47]:

1 *Threshold based updates*: This policy is based on a threshold (*tv*). If B^i_{adv} is the last advertised bandwidth of the link *i* and B^i_{real} is the current real bandwidth of that link, then an update message is triggered when

$$\left\| \frac{B^i_{adv} - B^i_{real}}{B^i_{adv}} \right\| > tv \qquad (1)$$

2 *Class based triggers*: Two examples are most representative, *Equal class based* updates and *Exponential class based* updates. Both are based on link partitioning. This partitioning is made either using a fixed bandwidth size *Bw* (*Equal class*), or adding to this value another constant value *f* (*Exponential class*). As a result, the total link capacity is divided into several equal size classes {*(0,Bw), (Bw, 2Bw),* . . .} or into several unequal size classes {(*0,Bw), (Bw, (f+1)Bw), ((f+1)Bw, (f*^{2}*+f+1)Bw),* . . .}. For these two policies, an update message will be sent only when the link capacity variation produces a change of class.

3 *Time based triggers*: In this case, updates are sent at fixed intervals. Also known as *hold-down timers*, these are usually implemented along with one of the above-described mechanisms, in order to reduce the number of update messages they generate.

In [48] a moving average technique is introduced as a substitute for the hold-down timers, with the objective of reducing the number of update messages. The authors show that by monitoring and filtering bandwidth utilization an average bandwidth utilization value can be computed, and this value can then be used to trigger update messages. Simulation results exhibit a more efficient reduction in signalling overhead when using this moving average technique.

In summary, in spite of the fact that the use of triggering policies to define when update messages must be sent reduces signalling overhead, it introduces a certain degree of routing uncertainty.

Some different sources of routing inaccuracy have been analysed so far. Now, the effects of this inaccuracy on the path selection process are described.

All routing algorithms rely on the accuracy of network state information to optimise the path selection process. When the information contained in the network state databases is not perfectly updated the routing process could potentially select a path that is unable to support the traffic requirements. In this case routing is done under inaccurate network state information; in other words, a certain degree of uncertainty exists in the routing information used to select the optimal path. As a consequence of this routing uncertainty, the connection blocking probability increases and the route selection process is definitely either badly performed or in the best case non-optimally performed. As it has been stated earlier, the problem of performing the routing decision under inaccurate routing information is known as the routing inaccuracy problem.

Having a certain degree of inaccuracy in the network state information is not of vital importance in traditional IP networks. This sentence must be correctly understood. The potential effects of having inaccuracy are the same in IP routing and in QoS routing. However, since in traditional IP routing only topological information, which does not suffer constant changes, is considered in the path selection process, usually link state databases may be updated by periodically (seconds or even minutes) sending update messages. Thus, because of the very low probability of changes in network topology, the degree of accuracy of the link state information is wholly dependent on the length of time between update messages. In other words, the risk of routing inaccuracy is very low.

A different situation occurs when more dynamic attributes are included in the network state information and changes in these parameters are often expected. This case appears when a certain QoS routing algorithm is implemented in a large, connection-oriented packet-switched network to select a path for a traffic flow requiring a certain QoS. In this case, since changes in the QoS parameters are

constantly expected, the number of update messages needed to keep the network state databases perfectly up-to-date is substantially greater. In fact, if the QoS parameter is the link available bandwidth, every time a connection is established or released this parameter will be modified. As a consequence, a large number of update messages flow throughout the network, which implies a non-desirable signalling overhead. The reduction of this overhead requires implementing additional mechanisms (i.e., triggering policies) in the routing protocol, which leads to an unfortunate consequence: network state information is not perfectly updated, so network state databases do not represent a current picture of the network.

In this case, the computations realized using inaccurate network state information could yield erroneous results. Subsequently, when this routing uncertainty is used to select the most suitable path through which traffic will be transported, it is possible that the selected route might not really have enough available resources to support the traffic requirements. Consequently, a certain incoming *LSP* demand that is initially allocated to a particular path would be rejected in the path set-up process, increasing the blocking probability.

At this point, it is important to note that the effects produced in the path selection vary depending on both the nature of the routing inaccuracy introduced and the QoS parameter required. First, the nature of the routing inaccuracy fundamentally affects the path selection process assuming that the inaccuracy is due to a triggering policy component. In this case, when the triggering policy in use does not include a hold-down timer it is possible to locate the current value of the QoS parameter into a range of values based on the last advertised value. This can be done because the details of the triggering policy's performance may be known allowing an analytical model to be generated to represent the routing inaccuracy effects. However, when the triggering policy includes a hold-down timer, since the time between two consecutive updates is usually large to reduce signalling overhead, the degree of inaccuracy is much larger and it is very difficult to make an appropriate estimation. These two triggering policy cases must be treated differently and different approaches must be implemented.

Second, the effects produced by routing inaccuracy in path selection will be different depending on the QoS parameter concerned: bottleneck requirements (bandwidth), or additive requirements (delay). Therefore, traffic flows with bandwidth requirements and traffic flows that impose an upper bound on the end-to-end delay should be handled differently. In fact, for traffic flows with bandwidth requirements, the optimal path could be found by extending existing standard routing algorithms with some modifications. For traffic flows with delay bounds, however, the problem is more difficult and several intractable issues appear when looking for the optimal path. Nevertheless, in both cases the objective is to find the optimal path that supports the incoming requirements considering that the routing information does not reliably represent the current network state. In summary, new routing algorithms, which account for routing inaccuracy in the path selection process, have to be sought.

This Thesis focuses on the routing inaccuracy problem when selecting paths with bandwidth requirements. A new routing mechanism is proposed to address the effects produced in global network performance because of having inaccurate routing information produced by using a certain triggering policy to reduce the signalling overhead needed to keep network state information perfectly updated in the edge nodes.

The network shown in Figure 3 illustrates the effects of selecting paths with bandwidth requirements under inaccurate network state information on the connection blocking probability.

Black link values (on the left) represent the residual available bandwidth of each link according to the network state information contained in the *TED*s on the edge nodes. Suppose that an incoming *LSP* demand reaches LSR1 demanding an *LSP* to LSR4 with 5 units of bandwidth. LSR1 selects two possible routes as shown in the Figure. Assume that for instance the shortest one is selected. Hence, a *Path* message is sent from LSR1 to LSR4 along LSR2 and LSR3, to set up the *LSP*. If there are enough resources a *Resv* message is sent back from LSR4 to LSR1 establishing the *LSP*. Now, traffic may be sent from LSR1 to LSR4 along the established *LSP*. Assuming that the updating mechanism is based on a certain triggering policy, it is

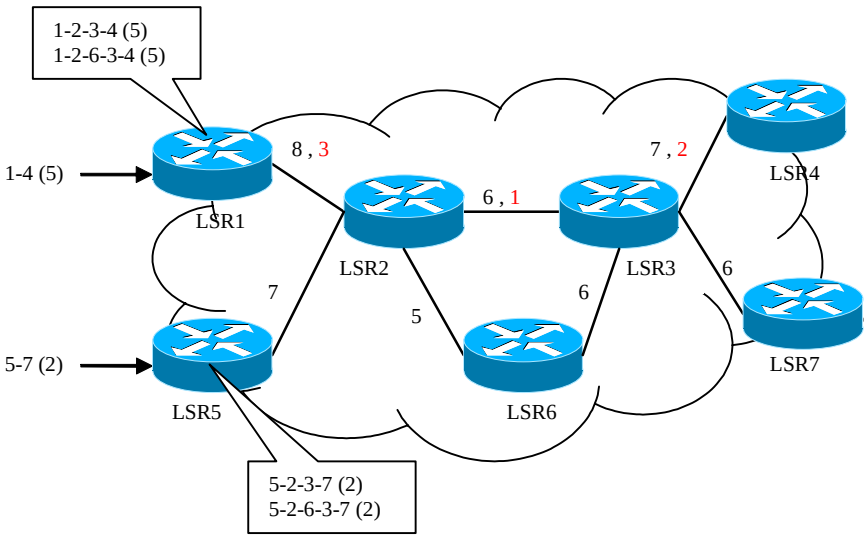


Figure 3. Routing inaccuracy effects in an IP/MPLS scenario

perfectly reasonable that update messages are not sent yet. Hence, in black (on the left) there are the last advertised residual bandwidth values of each link as read in the *TED*s and in red (on the right) there are the real bandwidth availability values once the first *LSP* has been established. Suppose that an incoming request reaches LSR5 demanding an *LSP* to LSR7 with 2 units of bandwidth. In this case the network state information from which LSR5 will select the path does not perfectly represent the real network state, which is the real bandwidth availability. LSR5 selects two possible routes and (as previously done) selects the shortest one. Hence, LSR5 sends a *Path* message from LSR5 to LSR7 through LSR2 and LSR3. When this *Path* message reaches LSR2 this node detects that there is not enough residual bandwidth on the output link to LSR3 to cope with the 2 units of required bandwidth. In this situation, LSR2 drops the *Path* message, so blocking the incoming *LSP* demand. The rejected *LSP* might be established if a routing algorithm considering inaccurate network state information was applied.

Chapter 5

Review of Existing Solutions

Several proposals dealing with finding a routing mechanism to improve routing performance under inaccurate routing information can be found in the literature. These works may be classified according to the following four approaches: (1) based on finding the path with the highest probability of having enough available resources to cope with the incoming requirements; (2) based on defining specific parameters, which reconcile the effects of having inaccurate routing information in the routing process; (3) based on improving the accuracy of network state information; and (4) based on addressing the routing inaccuracy effects introduced by the state aggregation in hierarchical networks.

Solutions following the first, probabilistic, approach include routing inaccuracy in the path selection process in order to find a path that guarantees the availability of enough resources to cope with the incoming requirements. The source node thus associates a certain stochastic behaviour to the QoS performance parameter, which is represented by a certain probability distribution function, so that an analytical model

incorporating this probability can be included in the routing process. It is important to realize that these probability distribution functions are not sent by the network nodes throughout the network as part of the link state advertisement: the source node builds these probability functions based on the received link state information.

Regarding the type of probability distribution function used, a first option would be to consider an exponential distribution, but due to the different sources of inaccuracy it would be very optimistic to suppose that this model would be useful for all the inaccuracy models. Finally, it is also important to take into account that the probability distribution function which models the routing inaccuracy can be artificially modified so that it includes any constraint desired to optimise the network performance (e.g. network load, paths successfully routed, etc.). A good example of routing algorithms based on this approach is *Safety Based Routing* [49], which is described later. Other significant contributions can be found in [50] and [51].

Examples of solutions following the second approach include *Ticket Based Probing* [52], a distributed multipath routing scheme based on probing multiple feasible paths simultaneously, and the *BYPASS Based Routing* mechanism introduced in this Thesis.

Solutions following the third approach lie in modifying the mechanism used to compute and utilize the link state attributes of each link in order to improve their accuracy. This set of solutions encompasses the works presented in [53] and [54] which are also described later on.

Finally, there are works that specifically deal with the effects that state aggregation introduces in the routing information. In this fourth group it is important to mention the crankback mechanism included in the *PNNI* protocol [46] to cope with the routing inaccuracy problem in an ATM scenario, which is presented in the final part of this Chapter. Other documents [55], [56], [57], [58], [59] present new aggregation techniques that reduce link state inaccuracy instead of substantially reducing the amount of data needed in the state.

Most significant routing mechanisms introduced in the literature fully thought to deal with the routing inaccuracy problem are now in detail described.

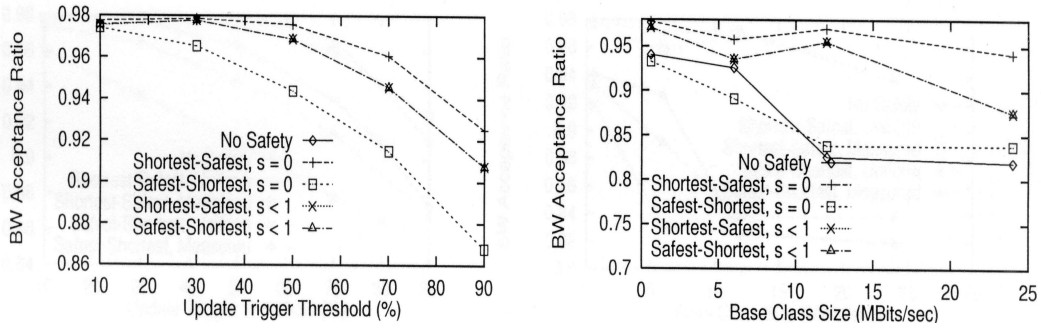
5.1 Safety Based Routing

Safety Based Routing (*SBR*), proposed by G.Apostopoulos et al in [49], assumes explicit routing with bandwidth constraints and on-demand path computation in such a way that a single path is computed when an incoming request reaches the source node. *SBR* is based on computing the probability that a path can support an incoming bandwidth request. This value, named safety (*S*), gives a measure of the probability that the total bandwidth required (b_{req}) is indeed available on the sequence of links composing each path (i.e., identifying those links having a higher probability of supporting the incoming request). This probability can be then used to classify every link, and obviously to find the safest path: the path having the best chance of supporting b_{req}.

Since the safety of each link is considered regardless of the other links in a path, the *safety* of a path (*S*) is computed as the product of every link present in that path. Once *S* has been computed it is included in the path selection process as a new parameter to be taken into account; thereby including a component of routing inaccuracy in the route selection process. As a result, the path that has larger probability *S* of having enough available capacity to provide the required bandwidth is selected.

Two different routing algorithms based on combining *S* with the number of hops are inferred from the *SBR* mechanism, namely the *Safest-Shortest Path* and the *Shortest-Safest Path*, the shortest-path being that path with the minimum number of hops. The *Safest-Shortest Path* algorithm selects that path with the largest *safety* among the shortest paths, while the *Shortest-Safest Path* algorithm selects the paths with largest *safety* and if more than one exists the shortest one is chosen.

To test these algorithms, in [49] a performance evaluation study is done by simulation. From this study it results that in terms of blocking probability, the *Shortest-Safest Path* (*SSP*) algorithm is the most effective for any of the triggering policies that were evaluated as shown in Figure 4.



(a) (b)

Figure 4. Bandwidth acceptance ratio in *SBR*: (a) Threshold; (b) Exponential classes

5.2 Explicit QoS Routing under Inaccurate Network State Information

This is one of the most known works published by R.Guerin and A.Orda in [50]. This work addresses the problem of selecting guaranteed QoS explicit paths under inaccurate network state information by selecting that path which has the largest probability of supporting the QoS incoming requirements. As explained earlier, the problem of selecting a path for traffic flows with bandwidth requirements and the same problem applied to traffic flows requiring end-to-end delay bounds must be separately solved, since they present a completely different degree of complexity. In this paper both the bandwidth constraint and the delay constraint are analysed.

Concerning to the bandwidth constraint the problem is to compute paths for those traffic flows with bandwidth requirements when the link state information contained in the source node can only be considered an inaccurate estimation of the bandwidth that is really available in the network components. This problem, known as *Most-Probable Bandwidth Constrained Path* (*MP-BCP*) can easily be solved by using a standard version of the *Most Reliable Path* (*MRP*) algorithm, which is based on computing the shortest paths for properly selected link weights. This solution efficiently addresses the problem of selecting paths for traffic flows with bandwidth requirements, but is not a good method for traffic flows requiring guaranteed end-to-end delay bounds, known as *Most-Probable Delay Constrained Path* (*MP-DCP*). In order to make the routing inaccuracy problem tractable when the objective is to guarantee that the end-to-end delay must be within a certain fixed delay threshold, the authors assume several approaches and generate two different models.

First, a "*rate based*" model is introduced. This model achieves the delay bound by ensuring a minimum service rate to the traffic flow. The main advantage of this model is that once the delay is mathematically represented, it is noticed that the end-to-end delay bound only depends on the available bandwidth on each link. Although this case and the bandwidth constraint case can be similarly addressed, they are not exactly the same, since the accumulative effect associated with delay is not produced in the bandwidth scenario. It is shown that the problem is intractable, although some solutions are presented for particular cases. Assuming that the obtained solution is not optimal but near-optimal, an algorithm named *QP* is introduced. However, this approach has the disadvantage that some constraints must be added, mostly regarding using schedulers that allow rates to be strictly guaranteed.

The second model is the "*delay-based*" model, which tries to guarantee a global delay in the selected path by concatenating the local delays associated with each node/link along the path. Therefore, in order to determine the end-to-end delay bounds, the information to be flooded throughout the network must be modified to include local delay, in such a way that the link state databases contain local delay information. Depending on the reliability of this information, the path selection process can be more or less affected, which explains why path computation is much more difficult in this model than in the *rate-based* model. Once the problem is mathematically defined in the paper, the authors conclude that although they have found tractable solutions for some cases, these cases are relatively limited; thus it is desirable to find a tractable general solution that can cope with most network conditions. In order to achieve this authors present a simplification, which reduces one general end-to-end delay guarantee problem to several minor problems. In fact the simplification is based on splitting the end-to-end constraint to a local delay constraint on each link.

5.3 QoS Routing with Uncertain Parameters for End-to-End Delay Constrained Traffic

Two different constraints, bandwidth and end-to-end delay, are analysed in the paper presented by D.Lorenz and A.Orda in [51]. The first constraint is defined as polynomial solvable, and the second is defined as computationally intractable. The

paper focuses on traffic with end-to-end delay requirements, providing new solutions and analysing their viability.

As previously defined, when end-to-end delay is the traffic flow QoS constraint, the inaccuracy is primarily due to the delay on each link, in such a way that a stochastic behaviour must be assigned as a parameter that represents the uncertainty. The paper considers a uniform distribution of the delay around the last advertised value, assuming the range to be defined by the threshold value used in the update process. However, the paper does not consider the exact mechanism used to obtain the delay probability distribution function. Hence, the main objective is to find the path most likely to cope with the delay constraint, which is named the *MP Problem*.

The complexity of the *MP Problem* depends on how the end-to-end delay constraint is split into several local end-to-end delay constraints. This process is not simple and the optimal partition is not easy to find. Hence, the initial *MP Problem* is extended to the *OP-MP Problem*, the *Optimally Partitioned MP Problem*. The authors find a solution for the *MP Problem* and the solid contribution of this work is that the solution is efficient for a wide class of probability solutions. This is achieved by defining a particular family of probability distributions (including normal and exponential distributions), where the family selection criterion is based on having a certain convexity property. Specifically, the paper extends an enhanced solution for a particular component of this family of distributions, the uniform distribution, generating an alternative algorithm for addressing the *OP Problem*, which is described in [50].

Once the authors present some solutions to deal with the *Optimal Partition* (*OP) Problem*, they go on to analyse the *OP-MP Problem*. The solution presented is based on using dynamic programming methods. In fact, the solution uses a modification of the *Dynamic-Restricted Shortest Path Problem* (*D-RSP*). The *RSP* problem is a well-known problem that aims to find the optimal path that minimizes the cost parameter among all the paths that satisfy the end-to-end delay constraint. Although the *RSP Problem* is NP-hard [60] authors propose a pseudopolynomial solution from which a new algorithm, *Dynamic-OP-MP*, is inferred. The main difference between the *Dynamic-OP-MP* algorithm and the *D-RSP* algorithm is the cost computation

method. As in the *OP Problem*, the *MP-OP Problem* is analysed in detail when a uniform distribution exists, yielding the *Uniform-OP-MP* algorithm.

Finally, an approach to obtaining a fully polynomial solution to deal with the *OP-MP Problem* is proposed. As in the last case, this approach is based on making some modifications to the *D_RSP* algorithm, such that the result is a non-optimal approximation (discrete solution) that introduces a bounded difference in terms of cost and success probability regarding the optimal solution. Basically, this solution interchanges the cost and delay roles in the algorithm.

5.4 Ticket-Based Distributed QoS Routing Scheme

The *Ticket-Based Distributed* QoS Routing mechanism was proposed by Chen and Nahrstedt in [52]. The authors focus on the NP-complete delay-constrained least-cost routing, and generate a routing algorithm designed to find the low-cost path, in terms of satisfying the delay constraint, by using only the available inaccurate or imprecise routing information. In order to achieve this purpose, the authors initially suggest a simple imprecise state model that defines which information must be stored in every node: connectivity information, delay information, cost information, and an additional state variable. Named delay variation, this additional state variable represents the estimated maximum change of the delay information before receiving the next update message.

It has to be noted that the imprecise model is not applied to connectivity and cost information, in order to simplify the process. The authors justify this assumption by saying that the global routing performance is not significantly degraded.

Then the authors propose a multipath distributed routing scheme, *ticket based probing*. *Ticket based probing* sends routing messages, *probes*, which are routed from a source *s* to a destination *d* according to the (imprecise) network state information available at the intermediate nodes, in order to find a low-cost path that fulfils the delay requirements of the *LSP* request. Each *probe* carries at least one ticket in such a way that by limiting the number of tickets, the number of probes is limited as well. Moreover, since each probe searches a particular path, the number of searched paths is also limited by the number of tickets. In this way, the trade-off

between signalling overhead and global routing performance may be controlled. Finally, based on this ticket based probing scheme, the authors suggest a routing algorithm to address the NP-complete delay-constrained least-cost routing problem, the *Ticket Based Probing* algorithm (*TBP*). Three algorithms are simulated in [52]: the flooding algorithm, the *TBP* algorithm and the shortest-path algorithm. Simulation results are represented using three parameters, the success ratio, the average message overhead, and the average path cost. The results obtained show that the TBP algorithm exhibits a high success ratio and a low-cost path, satisfying the delay constraint with minor overhead, and tolerating a high degree of inaccuracy in network state information.

5.5 Centralized Server Based QoS Routing

Unlike previous solutions, the solution proposed in [53] does not attempt to enhance the routing process assuming inaccurate network state information; rather, it seeks to eliminate the inaccuracy. In fact, the authors propose centralized server based QoS routing schemes which lead to both the elimination of the overhead needed to exchange network state update messages and the achievement of higher routing performance by utilizing accurate network state information in the path selection process. In this scenario, routing is handled by a route server and several

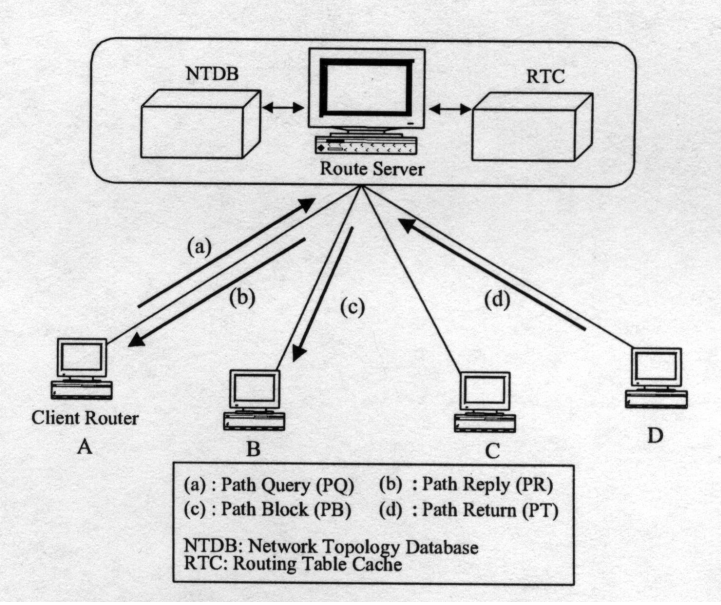


Figure 5. Centralized Server based QoS routing architecture

routers acting as clients: client routers send routing queries for each incoming request to the route server as shown in Figure 5. The route server stores and maintains two data structures: the *Network Topology Data Base* (*NTDB*), which contains link state information for each link in the network, and the *Routing Table Cache* (*RTC*), which stores the computed path information.

Although the main idea is derived from the mechanism suggested in [61], these new schemes use a different network state information collection method. Instead of collecting the link state information from the other routers, in this new approach the proposed router server updates and maintains a link state database as the paths assigned to or returned back from a certain flow. The main aspects to be considered in this centralized scheme are: (1) the processing load and storage overhead required at the server; (2) the protocol overheads to exchange the router queries and the replies between the server and the remote routers that act as clients; and (3) the effects produced when the server becomes either a bottleneck point or a single point of failure. The authors suggest various alternatives to reduce the loads and overheads, but leave the last aspect for future work.

There are two algorithms used to compute the path: a modification of Dijkstra's algorithm, and the Bellman-Ford algorithm with QoS extensions. Thus, assuming the existence of a certain locality in the communication pattern, a large number of source-destination pairs are expected to be unused. For this reason the authors use a path caching approach to reduce the path computation overhead. There are two parameters that limit the size of the *RTC*: *K*, defined as the maximum number of entries in the *RTC* (source-destination pairs), and *n*, defined as the maximum number of paths for each source-destination pair.

The server based QoS routing schemes introduced in this document are evaluated by simulation. Two types of results are obtained, one to show the effects of these schemes on path computation overhead, and the other to compare the proposed schemes with other QoS distributed routing schemes. The simulation results show that a simple path-caching scheme substantially reduces path computation overhead when considering locality in the communication pattern. Furthermore, according to

the simulation results the proposed schemes perform better than distributed QoS routing schemes while keeping a similar overhead.

5.6 A localized QoS Routing Approach

Document [54] focuses on localized QoS routing. The main advantage of using localized approaches for QoS routing is that no global network state information exchange among network nodes is needed, hence reducing the signalling overhead. In fact, path selection is performed in the sources nodes according to their local view of the global network state. However, the main difficult in implementing any localized QoS routing scheme is how the path selection is done based only on the local network state information collected in the source nodes. In order to address this problem the authors present a new adaptive proportional routing approach for designing localized QoS routing schemes. This approach provides an idealized proportional routing model, where all paths between a source-destination pair are disjoint and their bottleneck link capacities are known. However, this situation is not usually the case, and to address the effects produced by this deviation from reality, the concept of the *virtual capacity* of a path is introduced. The *virtual capacity* concept provides a mathematically sound way to deal with links shared among multiple paths. Based on this concept a theoretical scheme, *Virtual Capacity based Routing* (*VCR*), is described. Simulation results obtained show how the *VCR* scheme adapts to traffic load changes by adjusting traffic flows to the set of predefined alternative paths. However, the authors describe two significant difficulties related to *VCR* implementation, which suggest a new, easily realizable implementation of the *VCR* scheme: *Proportional Sticky Routing* (*PSR*). The *PSR* scheme can be viewed as operate in two stages: proportional flow routing and computation of flow proportions.

PSR proceeds in cycles of variable lengths. During each cycle, any incoming request can be routed along a certain path selected among a set of eligible paths, which initially may include all candidate paths. A candidate path becomes ineligible depending on the maximum permissible flow blocking parameter, which determines how many times this candidate path can block a request before becoming ineligible. When all candidate paths become ineligible a cycle ends and all the parameters are

reset to start the next cycle. An eligible path is finally selected to route the traffic depending on its flow proportion: the larger the flow proportion, the larger the chances of being selected. Simulation results show that the PSR scheme is simple, stable and adaptive, and the authors conclude that it is a good alternative to global QoS routing schemes.

5.7 Crankback and Fast Rerouting

The crankback and fast rerouting mechanisms were included in the ATMF *PNNI* [46] to address the routing inaccuracy problem due to fast changes in resource availability, and due to information condensation in a hierarchical network structure.

Assuming that a hierarchical network is divided into different peer groups, the mechanism is based on sending a *Release* message in the reverse path from the node that cannot forward the set-up message (i.e., the node that detects the set-up message blocking) to a node able to compute an alternate path in the same Crankback Level (i.e., the peer group level). The *Release* message contains a *Crankback Information Element* (*IE*), which indicates the blocked link and the *Crankback Level*. When a node in the reverse path receives a *Release* message it may either compute an alternate path or further crankback the message. This decision depends on alternate path availability and the routing information contained in that node. In this way, blocked links can be avoided by using an alternate path. When the crankback process returns back to the source node demanding the connection and this node also fails to find a path to the destination, the connection request is blocked or rejected. Therefore, the crankback mechanism does not guarantee a successful connection set-up, while consuming both much CPU-time in the nodes as control data.

This mechanism can be implemented over both a datagram network, where path selection and fast rerouting are dynamically performed by the nodes; and an explicitly routed network, where the selected path and the alternate routes are computed by the source nodes.

Chapter 6

The BYPASS Based Routing Mechanism

In this Chapter we describe a new QoS routing mechanism, *BYPASS Based Routing* (*BBR*) [62], which aims to reduce the effects of the routing inaccuracy problem when considering bottleneck requirements, the increase of the bandwidth blocking ratio and non-optimal path selection, in an *IP/MPLS* scenario. The main target is to obtain a routing mechanism that improves the behaviour obtained when applying other existing solutions. Assuming that the (*Safety Based Routing*) *SBR* [49] is the best solution to address the effects produced in global network performance when performing the path selection process under inaccurate available bandwidth information, the *BBR* mechanism is proposed as a new solution to improve *SBR*'s behaviour. The main concept of the *BBR* mechanism is the dynamic bypass concept, which is based on computing more than one feasible route to reach the destination. *BBR* instructs the ingress node to compute both the working route and a number of *bypass-paths:* paths that bypass those links that potentially might not be able to cope

with the incoming traffic requirements. Nevertheless, as we discuss below, only those paths that bypass links that truly lack enough available bandwidth to support the required bandwidth are set up.

Note that the idea of the *BBR* mechanism is derived from protection switching for fast rerouting, discussed in [63]. However, unlike the use of protection switching for fast rerouting, in this proposal both the working and the alternative path (*bypass-path*) are simultaneously computed but not set up. Alternative paths are only set up when required.

There are three main components in the *BBR* mechanism: to decide which links should be bypassed, to compute the *bypass-paths*, and to decide when *bypass-paths* must be used. These three components may be summarized as follows:

1) ***Obstruct-Sensitive Links***: A new policy must be added in order to find those links (*Obstruct-Sensitive Links*, *OSL*s) that might not be able to support the traffic requirements associated with an incoming *LSP* demand. This policy must guarantee that whenever a set-up message sent along the explicit route reaches a link that does not have enough residual bandwidth to support the required bandwidth, this link had previously been defined as an *OSL*.

2) ***Working path selection*:** Using *BBR*, two different routing algorithms can be initially analysed. These algorithms are obtained from the combination of the Dijkstra's algorithm (in terms of hopcount) and the *BBR* mechanism. Therefore, two different strategies may be applied:

   - The *Shortest-Obstruct-Sensitive Path* (*SOSP*), computing the shortest path among all the paths that have the minimum number of *Obstruct-Sensitive Links*.
   - The *Obstruct-Sensitive-Shortest Path* (*OSSP*), computing the path that minimizes the number of *Obstruct-Sensitive Links* among all the shortest paths.

3) ***Bypass-paths*, calculation and utilization:** Once the working path is selected the *BBR* computes the *bypass-paths* needed to bypass those links in the working path defined as *OSL*. When the working path and the *bypass-paths* are computed, the working path set-up process starts. Thus, a signalling

message is sent along the network following the explicit route included in the set-up message. When a node detects that the link through which the traffic must flow does not have enough available bandwidth to support the required bandwidth, it sends the set-up signalling message along the *bypass-path* that bypasses this link. Thus, the set of bypassed links must always be defined as *OSLs* so that a feasible *bypass-path* exists. Moreover, it is important to note that the *bypass-path* nodes are included in the set-up signalling message as well (i.e., *bypass-paths* are also explicitly routed). A possible option to implement the mechanism used to set up the *bypass-paths* is described later in this Chapter. In order to minimize the set-up message size, *bypass-paths* are removed from the set-up message when the links that they are intended to bypass have been traversed.

6.1 Description of BYPASS Based Routing

Let *G(N,L,B)* describe a defined network, where *N* is the set of nodes, *L* the set of links and *B* the bandwidth capacity of the links. Suppose that a set *P* of source-destination pairs *(s,d)* exists, and that all the *LSP* requests occur between elements of *P*. Let b_{req} be the bandwidth requested in an element *(s,d)* $\hat{I}$ *P*.

***Rule 1*:** Let $G_r(N_r,L_r,B_r)$ represent the last advertised residual graph, where N_r, L_r and B_r are the remaining nodes, links and residual bandwidths respectively at the time of path set-up. Let L^{os} be the set of *OSLs* (l^{os}), where the elements of L^{os} are determined depending on the triggering policy in use. Therefore,

- Threshold policy: Let b^i_r be the last advertised residual bandwidth for a link l_i, and let *tv* be the threshold value. This link l_i is defined as an *OSL*, l^{os}_i if

$$l_i = l^{os}_i \mid l^{os}_i \in L^{os} \Leftrightarrow b_{req} \in (b^i_r(1\text{-}tv), b^i_r(1+tv)] \; . \qquad (2a)$$

- Exponential class policy: Let $B^i_{l\_j}$ and $B^i_{u\_j}$ be the minimum and the maximum bandwidth values allocated to class *j* for a link l_i. So, l_i is an *OSL*, l^{os}_i if

$$l_i = l^{os}{}_i \mid l^{os}{}_i \in L^{os} \Leftrightarrow b_{req} \in (B^i{}_{l\_j}, B^i{}_{u\_j}] \,. \qquad (2b)$$

Rule 2: Let L^{os} be the set of *OSLs*. Let i_j and e_j be the edge nodes of a link $l^{os}{}_j$ $\hat{I}$ L^{os}. Let l_k be a link adjacent to $l^{os}{}_j$. The edge nodes of the *bypass-paths* to be computed are

$$(i_j, e_j) \qquad \Leftrightarrow \quad l_k \notin L^{os} \qquad (3a)$$

or

$$(i_j, e_k) \text{ and } (i_k, e_k) \qquad \Leftrightarrow \quad l_k = l^{os}{}_k \in L^{os} \,. \qquad (3b)$$

In this way two or more adjacent *OSLs* could be bypassed by a single *bypass-path*, as shown in Figure 6.

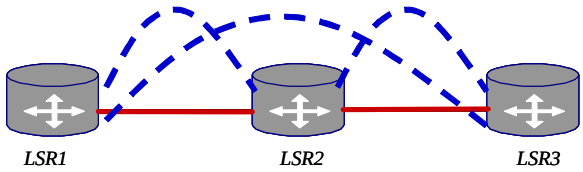


Figure 6. Rule 2 illustration

Three different situations exist: (1) Link LSR1-LSR2 is defined as an *OSL* but the adjacent link LSR2-LSR3 is not an *OSL*. In this case, if LSR1 detects that the requested bandwidth is unavoidable on the output link, forwards the set-up message along the *bypass-path* to LSR2; (2) Link LSR1-LSR2 is not defined as an *OSL* but link LSR2-LSR3 does. In this case, the *bypass-path* from LSR2 to LSR3 is used to send the set-up message when is needed; (3) Link LSR1-LSR2 and link LSR2-LSR3 are defined as *OSL*s. In this situation if LSR1 detects that requested bandwidth is unavoidable on the link of the primary path, it sends the set-up message along the *bypass-path* from LSR1 to LSR3 instead of LSR1-LSR2.

In accordance with these rules, a brief description of the *BBR* mechanism is presented in Figure 7. Steps 4 and 5 should be explained in detail. Once a link is defined as an *OSL*, the *BBR* mechanism computes the *bypass-path* that bypasses this link. The *bypass-paths* are computed according to *SOSP*, the shortest path among those paths minimizing the number of *OSLs*. Other criteria could be used to select the

BYPASS BASED ROUTING (BBR)

Input*:* The input graph $G_r(N_r,L_r,B_r)$. The *LSP* request is between a source-destination pair *(s,d)* and the bandwidth requirement is b_{req}.

Output: A route from *s* to *d* with enough *bypass-paths* to bypass the routing inaccuracy effects in the *obstruct-sensitive links.*

Algorithm:

1. Mark those links that are defined as *obstruct-sensitive link (OSL)* according to Rule 1.
2. Depend on the algorithm to be used, *OSSP, SOSP:*

   SOSP *(shortest-obstruct-sensitive path)*:
   - Compute the weight of a link *l* as
     $$w(l) = 1 \Leftrightarrow l \in L^{os}, \qquad w(l) = 0 \Leftrightarrow l \notin L^{os}$$
   - Apply Dijkstra's algorithm to select the paths $p \hat{I} P$ that minimize the number of *OSLs* by using *w(l)* as the cost of each link
   - If more than one path *p* exists selects the path that minimizes the number of hops

   OSSP *(obstruct-sensitive-shortest path)*:
   - Apply Dijkstra's algorithm to select the paths $p \hat{I} P$ that minimize the number of hops by using *w(l)* =1 as the cost of each link.
   - If more than one path *p* exists selects the path that minimizes the *OSLs.*
3. Compute a *bypass-path* for all the *OSLs* included in the selected path according to Rule 2.
4. Decide which *bypass- paths* must be used in accordance with real available resources in the path setup time.
5. Route the traffic from *s* to *d* along the setup path.

Figure 7. BYPASS Based Routing mechanism

bypass-paths, such as to simply apply *OSSP* or to maximize the residual available bandwidth. These different approaches are left for further studies

There are two main factors contributing to the complexity cost of *BBR*. First, selecting the shortest path by using a binary-heap implementation of the Dijkstra algorithm introduces a cost of $O(L \cdot \log N)$. Second, additional cost is introduced by the *bypass-path* computation. Assuming that the *bypass-path* cannot include a network element which is also included in the working path, G(V, E) represents the reduced network, where $V < N$ and $E < L$. Hence, a factor of $O(E \cdot \log V)$is added in order to compute one *bypass-path*. In addition, since a variable number M of *bypass-paths* may be computed along a working path, the cost is $O(M(E \cdot \log V))$. However, $\hat{M}$ being an upper bound of the number of computed *bypass-paths* along a working path, the complexity reduces to $O(\hat{M}(E \cdot \log V))$: effectively to $O(E \cdot \log V)$. Hence, the complexity is $O(L \cdot \log N) + O(E \cdot \log V)$. This expression may be finally reduced, considering that the *bypass-paths* are computed based on a reduced graph. Therefore, the complexity is $O(L \cdot \log N)$.

6.2 *Bypass-path* Signalling

At this point a new solution for signalling *bypass-paths* is introduced. This solution is borrowed from [64], where a solution for establishing and signalling alternative paths for fast rerouting in *IP/MPLS* networks is suggested. Assuming that the *Resource Reservation Protocol* (*RSVP*) is used as the *Label Distribution Protocol* (*LDP*), there are two main messages (*Path* and *Resv)* involved in the path set-up process. Hence, while the *Path* message flows downstream from the source node to the destination node requesting the label allocation, the *Resv* message flows generally upstream from the destination node to the source node establishing the path by allocating the label on each node along the selected path. A new object containing the intermediate node addresses along the path (the *Explicit Routing Object* (*ERO*) [13]) is added to the *Path* message when using explicit routing. The *ERO* is unique and will be made up of a set of sub-objects with the addresses that the *Path* message must follow to reach the destination node.

The *BBR* mechanism assumes that the source node detects *OSL*s and then a *bypass-path* route is computed for each one of these links. These *bypass-paths* are then sent along with the working path in the *Path* message, and each intermediate node selects which route use for the incoming *Path* message depending on its real resource availability. Since only those links defined as *OSL*s must be associated with a *bypass-path*, a new field is added to the *ERO* sub-objects to signify an *OSL*, and a *bypass-path* should be explicitly signalled for this link as well. In practice, the node upstream of the link defined as an *OSL* is the node where the link is marked as an *OSL*. This new field, named *Protect* (P) to maintain the nomenclature used in [64], consists of one bit that is set ‘true’ when the output link is an *OSL*. Therefore, we can say that a node is *protected* when the output link on which the traffic must flow has been identified as an *OSL*. In Figure 8 the format of an IPv4 address for a sub-object of the *ERO* including the *Protect* field is shown.

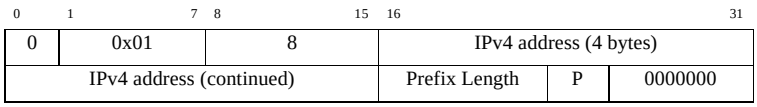


Figure 8. IPv4 sub-object of the *ERO*

Once a link is defined as an *OSL*, the *Path* message must carry the computed *bypass-path* that could be used by the protected node to bypass the link when there are not enough resources to cope with the traffic requirements. The addresses of the intermediate nodes on this *bypass-path* must also be explicitly piggybacked in the *Path* message. Therefore, the sub-objects of the *ERO* must be again modified, adding the new *Bypass-Path Address* (*BPA*) field. It is a variable length field, always a multiple of 32 bytes, which is only useful when the node has been defined as protected. In this case, the *BPA* field provides the *n* addresses that make up the *bypass-path*, from the protected node to another node along the working path, according to the routing algorithm's decision. The *BPA* field addition is shown in Figure 9. Two cases are depicted: in (a) the node that receives the *ERO* (@IPv4-i) is not a protected node (*P* = 0), so the output link where the traffic will be sent is not an *OSL*; in (b) the node is protected (*P* = 1) and the *BPA* field is shown. In this case the *BPA* consists of two IPv4 addresses (@IPv4-1 and @IPv4-2), which explicitly configures the *bypass-path* that bypasses the link directly connected to this protected node.

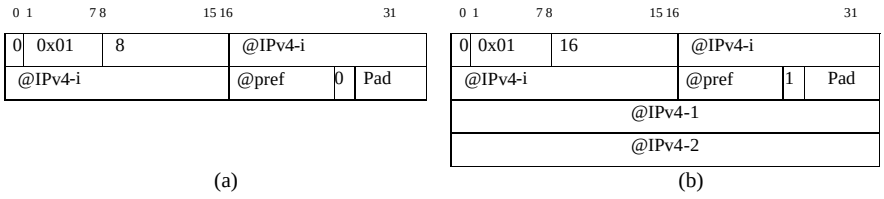


Figure 9. IPv4 sub-object of the *ERO*: (a) non-protected node; (b) protected node

It is important to note that a different situation may be found. In fact, even though a node is defined as protected, the existence of a *bypass-path* is not always guaranteed. This case matches the scenario in which there is not another feasible route between the edge nodes of an *OSL*. In this situation, the *P* field is set but the *length* field is left unchanged. This weakness of the *BBR* mechanism is addressed in Chapter 8.

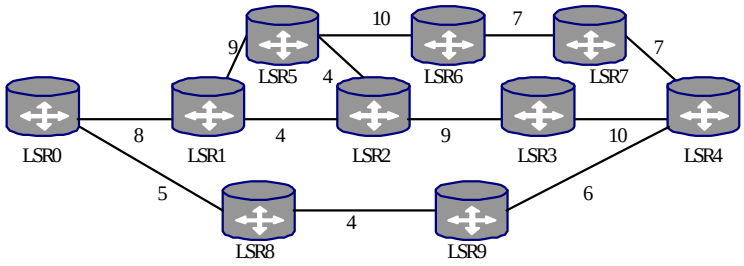


Figure 10. Network topology used to illustrate the *BBR* behaviour

6.3 Example for Illustrating BBR Behaviour

Before analysing the suggested algorithms in a large topology, we can test *BBR* performance in the simple topology shown in Figure 10, which shows the network topology, where the number associated with each link shows the residual units of bandwidth available. An *Exponential class* triggering policy is used, with $f = 2$ and $Bw = 1$ (as used in [49]), such that the resulting set of classes on each link are {(0,1], (1,3], (3,7], (7,15],…}. Moreover, an incoming *LSP* request is assumed to demand b_{req} of 4 units of bandwidth between LSR0 and LSR4.

In order to compare the *BBR* mechanism with other related work, the *Shortest-Safest Path (SSP)* algorithm presented in [49], which computes the path based on maximizing the "*probability of success*" of supporting the bandwidth requirements, is chosen as the sample routing algorithm. Thus, the algorithms tested in this example are *SSP*, *WSP* (as explained in Chapter 3, it selects the widest path among the shortest ones in the pruned graph which only contains links supporting the requested bandwidth), *SOSP*, and *OSSP*. The shortest path algorithm (SP) implemented in the OSPF routing protocol is used as a routing algorithm that does not consider routing inaccuracy when selecting the path. Table 2 describes the link QoS parameters used to compute the path, where B_t, *Class* and *S* are the bandwidth, class and *safety* associated with each link. The *S* value has been computed according to the expressions found in [49]. Note that *S* represents the probability that the requested amount of bandwidth is indeed available on a given link. Using this information the *BBR* mechanism is applied.

Table 2. Link QoS attributes

Link	B_t	*Class*	*S*
0-1	8	7,15	1
1-2	4	3,7	0,75
2-3	9	7,15	1
3-4	10	7,15	1

Link	B_t	*Class*	*S*
1-5	9	7,15	1
5-2	4	3,7	0,75
5-6	10	7,15	1
6-7	7	3,7	0,75

Link	B_t	*Class*	*S*
7-4	7	3,7	0,75
0-8	5	3,7	0,75
8-9	4	3,7	0,75
9-4	6	3,7	0,75

Table 3 shows different possible routes from LSR0 to LSR4, including the number of hops *H*, the number of *Obstruct-Sensitive Links OSL*, the minimum last advertised residual bandwidth b_r^{min}, and the *safety* parameter *S* for each path. Different paths are selected depending on the algorithm in use, as shown in Table 3. It is important to note that the *SOSP* and the *SSP* algorithms select the same route. Assuming that this is the normal *SOSP* behaviour, the blocking probability obtained by both algorithms will be the same. However, although the *SOSP* and the *SSP* algorithms select the same route, the key difference between the algorithms is the use of *bypass-paths* when needed. In fact, the *OSL* definition is a different manner of representing the *safety* of a link, which provides rerouting capabilities to some intermediate nodes. Hence, *bypass-path* utilization will reduce the blocking probability. Moreover, the *OSSP* algorithm exhibits a different behaviour (i.e., selects a different path) in the example. No prediction about bandwidth blocking results in comparison with the *SSP* algorithm can be made in advance, due to the possibility of *bypass-path* utilization.

Table 3. Feasible routes and selected paths depending on the algorithm in use

Id	*Route (LSR)*	*H*	*OSL*	b_r^{min}	*S*
a	0-1-2-3-4	4	1	4	0.75
b	0-1-5-6-7-4	5	2	7	0.56
c	0-1-5-2-3-4	5	1	4	0.56
d	0-8-9-4	3	3	4	0.42

Alg	*Path*
SP	d
WSP	d
OSSP	d
SOSP	a
SSP	a

Once feasible routes have been computed, the *bypass-path* selection process starts. If the *SOSP* algorithm is in use there is only one *OSL* in the route *a*, which can be bypassed by the path LSR1, LSR5 and LSR2. However, when the *OSSP* algorithm is in use, the process is much more complex since there are some *OSLs* that cannot be bypassed, e.g. link LSR8-LSR9. In this case *BBR* cannot be applied. A method for bypassing *OSLs* that do not have *bypass-paths* between their edge nodes

must be sought. One approach to address this case is to find different edge nodes for the *OSL* (the *BYPASS Discovery Process*), which is analysed in Chapter 8. Currently, as pointed out above, the *bypass-paths* are always computed by minimizing the number of *OSLs*.

Finally, after computing the *bypass-paths*, a path set-up message is sent along the working path. Each node checks the real available link bandwidth, and depending on this value the set-up message is sent through either the working path or the *bypass-path*.

6.4 Performance Evaluation

In this section we compare by simulation the *SOSP* and the *OSSP* algorithms inferred from the *BBR* mechanism with the *WSP* and the *SSP* algorithms. We exclude *SWP* due to its worse performance shown in [65]. The parameters used to measure the algorithms' behaviour are the routing inaccuracy and the blocking ratio.

- ***Routing Inaccuracy:*** This parameter represents the percentage of paths that have been incorrectly selected, defined as:

$$\text{routing inaccuracy} = \frac{\text{number of paths incorrectly selected}}{\text{total number of requested paths}} \qquad (4)$$

  A path can be incorrectly selected because of two factors. The first is an *LSP* request rejection when a route with enough resources is available to support that demand. The second factor is the blocking of an *LSP* that was initially routed by the ingress node but later rejected due to insufficient bandwidth in an intermediate link.

- ***Blocking Ratio*****:** We use the bandwidth-blocking ratio defined as:

$$\text{bandwidth blocking ratio} = \frac{\sum_{i \in \text{rej\_LSP}} \text{bandwidth}_i}{\sum_{i \in \text{tot\_LSP}} \text{bandwidth}_i} \qquad (5)$$

  where *rej_LSP* is the set of blocked demands, and *tot_LSP* is the set of all requested *LSP*s.

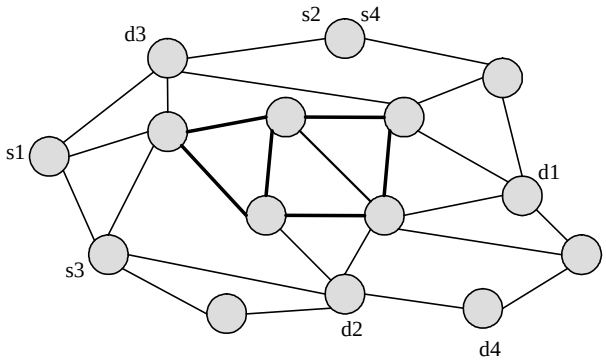


Figure 11. Network topology used in the simulations

The simulations are performed over the network topology shown in Figure 11, borrowed from [24], using the ns2 simulator extended with *MPLS* and *BBR* features. Two link capacities are used: 622 Mb/s, represented by a light line; and 2.5 Gb/s, represented by a dark line. The source nodes (s) and the destination nodes (d) are those shown in Figure 11. Every simulation requests 2000 *LSP*s from s_i to d_i, which arrive following a Poisson distribution, where the requested bandwidth is uniformly distributed between 1 Mb/s and 5 Mb/s. The holding time is randomly distributed with a mean of 60 sec. The *Threshold based triggering policy* and the *Exponential class based triggering policy* with $f = 2$, are implemented in the simulation. Let *n_bp* be the number of *bypass-paths* that may be computed per route. In order to reduce the computational cost, in these simulations three *bypass-paths* may be computed per route, so *n_bp* = 3.

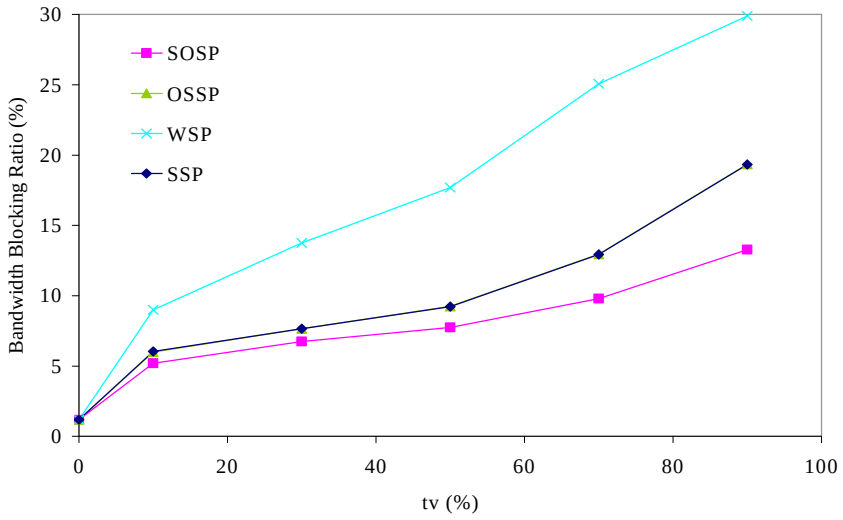


Figure 12. Bandwidth Blocking Ratio for the Threshold triggering policy

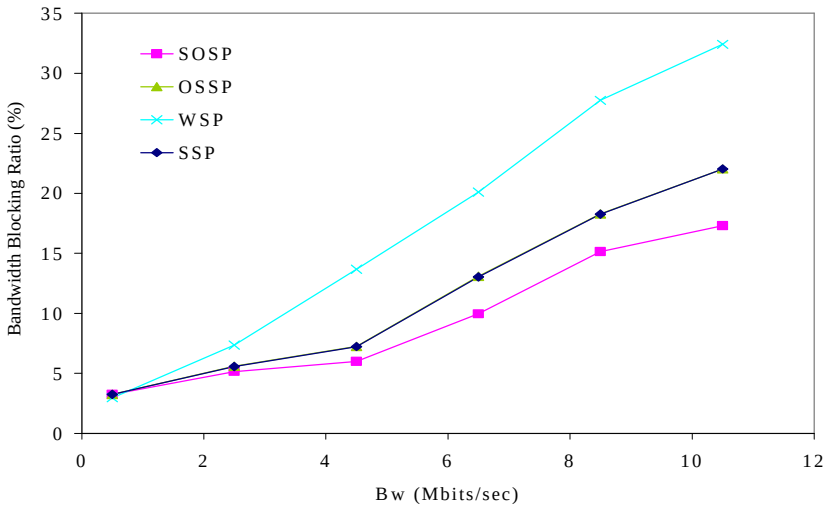


Figure 13. Bandwidth Blocking Ratio for the Exponential class triggering policy

All the results are obtained with a 95% confidence interval after repeating the experiment 10 times, with each simulation lasting 259 seconds. Figure 12 and Figure 13 show the bandwidth-blocking ratio for the *Threshold* and the *Exponential class* triggering policies respectively. The x-axis represents the threshold value *tv* and the base class size *Bw* for both triggering policies respectively. Remind that these values are used by the triggering policies to trigger the update messages.

The algorithms derived from the *BBR* mechanism (*OSSP* and *SOSP*) perform better than *WSP*. In addition, while *OSSP* yields similar results to *SSP*, *SOSP* is substantially better than the *SSP* performance. Specifically, for the *SOSP* algorithm

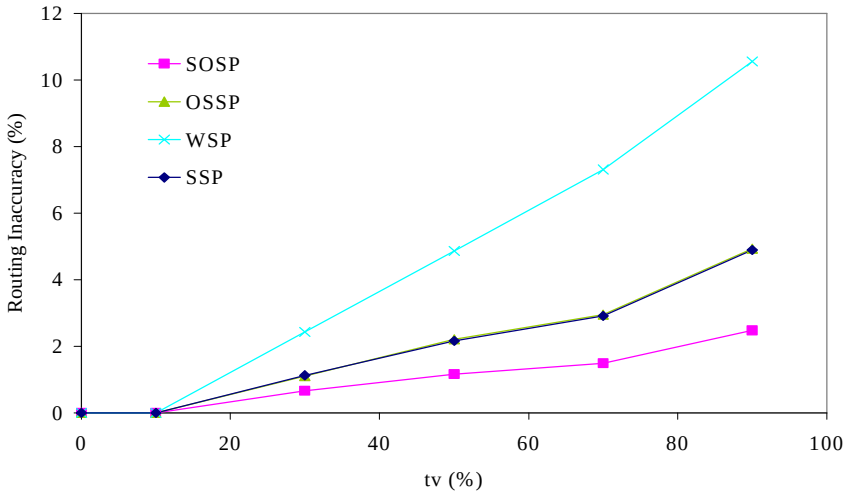


Figure 14. Routing Inaccuracy for the Threshold triggering policy

the Threshold value can be increased 10% while keeping the same bandwidth blocking ratio as *SSP*.

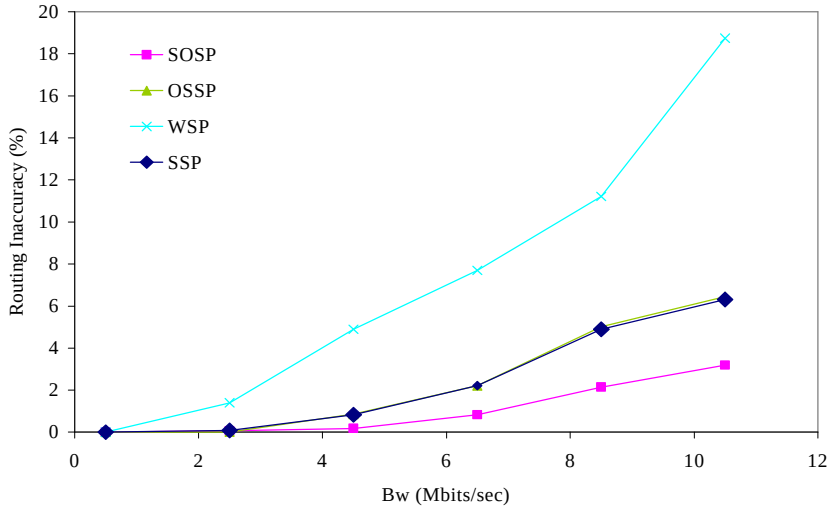


Figure 15. Routing Inaccuracy for the Exponential class triggering policy

Figure 14 and Figure 15 represent the routing inaccuracy for both triggering policies Threshold based and Exponential class based respectively. The *SOSP* algorithm presents the best behaviour, computing a lower number of incorrect routes.

The *OSSP* algorithm behaves better than the *WSP* algorithm although does not substantially improve the *SSP* behaviour.

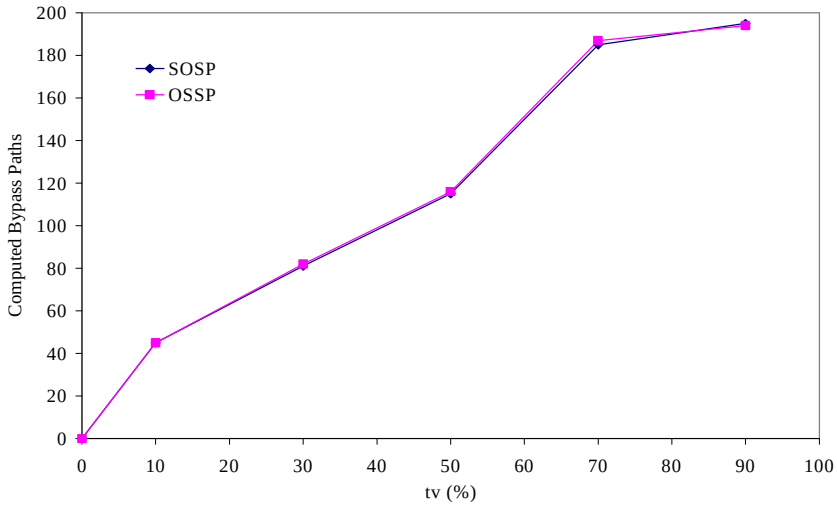


Figure 16. Computed *bypass-paths* for the Threshold triggering policy

The cost of the *BBR* mechanism for both triggering policies in terms of the number of computed *bypass-paths* is shown in Figure 16 and Figure 17. Remind that in order to reduce the computational cost in these simulations, *n_bp* = 3.

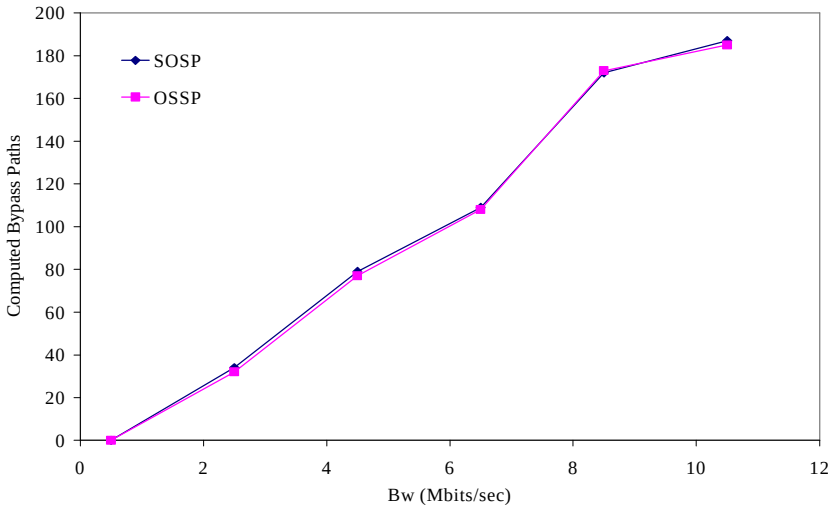


Figure 17. Computed *bypass-paths* for the Exponential class triggering policy

Both figures show that the cost is similar for both algorithms derived from the *BBR* mechanism. It reinforces the conclusion that *SOSP* behaves better than the *OSSP* algorithm. *SSP* and *WSP* do not incur the cost depicted in these figures. Note, however, that this cost is low given the benefits provided by the *BBR* mechanism.

In summary, as a numeric example we take a *tv* value of 70% and analyse the results provided by the *BBR* mechanism and those provided by the *SSP* algorithm, showing that the bandwidth blocking ratio presented by *SOSP* (9.7%) is substantially lower than that provided by *SSP* (12.9 %). Regarding routing inaccuracy, *SOSP* (1.49%) yields a lower number of incorrectly selected paths compared to *SSP* (2.91%). In both cases *OSSP* presents results similar to *SSP*, and *WSP* presents the worst behaviour. This is due to the fact that *WSP* does not consider routing inaccuracy when selecting a path. Finally, for *tv* = 70% the number of *bypass-paths* computed by the *BBR* mechanism during the simulation is close to 180 for both the *SOSP* and the *OSSP* algorithms. That means an *LSP* computation overhead of about 9%, but not in signalling, as explained earlier.

Chapter 7

Applying the BBR Mechanism under Bandwidth Constraints

In the previous Chapter the behaviour of the *BBR* was analysed and its benefits were also shown by simulation. Now, in this Chapter an enhancement of the *BBR* mechanism [66] to optimise bandwidth allocation is proposed. In fact, the algorithms inferred from the *BBR* mechanism presented so far do not explicitly include the available bandwidth in the path selection process; instead, they only consider a range of bandwidth values (those included in the *OSL* definition). The *BBR* enhancement presented here is based on balancing the path length and the residual bandwidth. *SOSP* being the best performing *BBR* algorithm, a simpler initial approach for selecting routes in accordance with certain bandwidth constraints is based on including the residual bandwidth in the *SOSP* algorithm. In this way *SOSP* is only modified when the final selection includes more than one path. In this case, the route is not randomly selected but the widest is chosen. We call this algorithm *Widest-Shortest-Obstruct-Sensitive-Path* (*WSOSP*).

WSOSP is just an initial approach, where the number of hops has more weight than bandwidth capacity in the route selection process. Therefore, in order to balance the path selection process, avoiding those paths that are both widest but too long and shortest but too narrow, a new algorithm is suggested. We call this algorithm *Balanced-Obstruct-Sensitive-Path* (*BOSP*). The *BOSP* algorithm is based on extending the shortest path algorithm with the number of *OSL*s, but unlike previous algorithms based on the *BBR* mechanism already described in the last Chapter, a new parameter is added to each feasible route between source and destination node pair. This parameter, F_p, represents the relation between the maximum residual bandwidth and the number of hops along a path $p \in P$:

$$F_p = n\left[\max\left(\frac{1}{b_r^i}\right)\right], \ldots\ldots\ldots i=1..n \qquad (6)$$

where n is the number of hops and b^i_r is the available residual bandwidth on link i in the path p.

In this way, by using F_p as the cost of each link, network load and network occupancy are balanced in the path selection process.

BALANCED OBSTRUCT SENSITIVE PATH ALGORITHM (BOSP)

Input: The input graph $G(N_r,L_r,B_r)$. The *LSP* request is between a source-destination pair *(s,d)* and the bandwidth requirement is b_{req}.

Output: An optimized and balanced route from *s* to *d* with enough *bypass-paths* to bypass the routing inaccuracy effects in the *obstruct-sensitive links*.

Algorithm:

1. Mark those links that are defined as *OSL* according to Rule 1
2. Compute the weight of a link *l* as
   $w(l)= 1 \Leftrightarrow l \in L^{os}, \qquad w(l)= 0 \Leftrightarrow l \notin L^{os}$
3. Apply Dijkstra's algorithm to select the path that minimizes the number of *OSLs* by using *w(l)* as the cost of each link
4. If more than one exists compute the cost F_p of each path
   $$F_p = n\left[\max\left(\frac{1}{b_r^i}\right)\right] \qquad i=1..n$$
5. Select the path that minimizes F_p
6. Determine the edge nodes pair *(i,e)* of the *OSLs* existent in the selected path
7. Compute the *bypass-paths* for each element *(i,e)* according to Rule 2
8. Decide which *bypass-paths* must be used in accordance with real available resources in the path setup time
9. Route the traffic from *s* to *d* along the setup path

Figure 18. *BOSP*: The enhanced *BBR* mechanism

In Figure 18 the *BOSP* algorithm is briefly described. Again, if more than one possible *bypass-path* exists, the route that minimizes the number of *OSL*s is chosen. The complexity of *BOSP* can be represented in the same manner used to represent the complexity of the initial algorithms inferred from the *BBR* mechanism, *SOSP* and *OSSP*.

As a summary, Figure 19 includes a flowchart depicting all the algorithms inferred from the *BBR* mechanism, i.e., the *SOSP*, *OSSP*, *WSOSP* and *BOSP* algorithms.

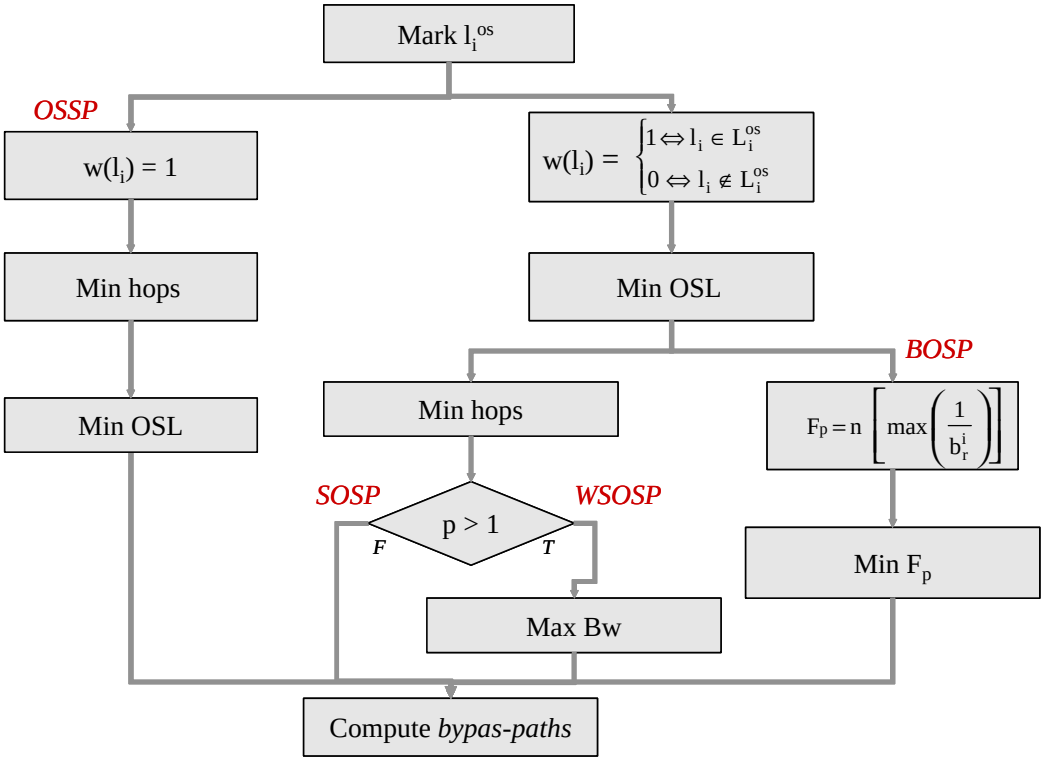


Figure 19. Routing algorithms inferred from the *BBR* mechanism

7.1 Example Illustrating the BOSP Behaviour

The topology shown in Figure 20 is used to test the performance of *BBR*. This test supposes an incoming *LSP* request demanding b_{req} of 4 units of bandwidth between LSR0-LSR7. Moreover, the triggering policy used is the Exponential class based policy, with $f = 2$ and $Bw = 1$ (as used in [49]).

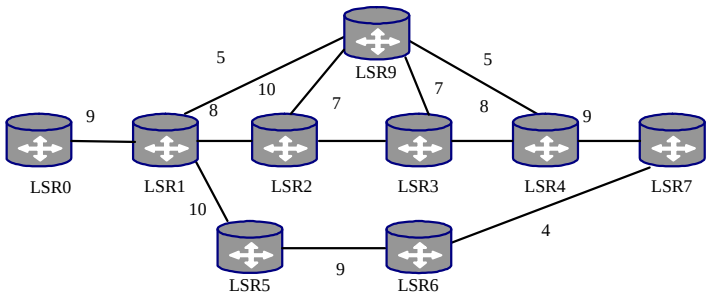


Figure 20. Network topology used to illustrate the *BOSP* algorithm

Table 4 shows all the different routes between *LSR0* and *LSR7*. *ID* is defined as a path identifier; *H* represents the number of hops along the path; b_r^{min} is the minimum residual bandwidth along the path; *N_OSL* is the number of *Obstruct-Sensitive Links*, F_p is the cost, and *Algorithm* represents the algorithms that select each path. The two new routing algorithms based on the *BBR* mechanism, *WSOSP* and *BOSP*, select the paths identified by *b* and *a* respectively.

Table 4. Link QoS attributes

ID	*PATH (LSR)*	*H*	b_r^{min}	*N_OSL*	F_p	*Algorithm*
a	0-1-2-3-4-7	5	7	1	0.71	BOSP
b	0-1-5-6-7	4	4	1	1	SOSP,WSOSP
c	0-1-2-9-3-4-7	6	7	1	0.85	
d	0-1-9-3-4-7	5	4	2	1.25	
e	0-1-9-2-3-4-7	6	4	2	1.5	
f	0-1-2-9-4-7	5	4	2	1.25	
g	0-1-9-4-7	4	5	2	0.8	WSP

Once the path has been selected, the *BBR* mechanism computes a *bypass* path for each *OSL*. When the *BOSP* routing algorithm is used, *a* is the route selected and one *OSL* exists, as shown in Table 4. Then, the *BBR* mechanism computes the edge nodes of said *OSL* that is {LSR2-LSR3}. In order to compute the *bypass-path*, it is possible to use any of the parameters shown in Table 5 (*H*, b_r^{min}, F and *N_OSL*). In this case, *N_OSL* is the parameter used to compute the *bypass-path*s. Therefore, {LSR2-LSR9-LSR3} is the *bypass-path* selected to bypass the *OSL*. When *WSOSP* is implemented, the edge nodes to bypass are {LSR6-LSR7}. However, in this case it is not possible to find a path that bypasses this link in the network topology. Hence, there are again some cases in which the dynamic bypass concept cannot be applied.

Table 5. Possible *Bypass-paths*

ID	L^{os}	*Edge LSRs*	*Bypass_path (LSR)*	*H*	b_r^{min}	F_p	*N_OSL*
a	2-3	2-3	2-9-3	2	7	0.28	1
b	6-7	6-7	--	-	--	--	--

7.2 Performance Evaluation

The ns/2 simulator used earlier, extended to implement the new algorithms, has been used to evaluate the performance of the proposed *BBR* enhancement. A set of simulations have been performed on different scenarios to evaluate the suggested *WSOSP* and *BOSP* algorithms in comparison with the already existing *WSP*, *SSP*, *SOSP* and *OSSP* algorithms. All the results have been obtained with a 95% confidence interval after repeating the experiment 10 times. The *BBR* mechanism has been evaluated in three different scenarios.

1) Scenario 1:

The simulations were carried out over the network topology shown in Figure 11. Every simulation requests 2000 *LSP*s, which arrive following a Poisson distribution where the requested bandwidth is uniformly distributed between 1 Mb and 5 Mb and the holding time is randomly distributed with a mean of 120 seconds. The triggering

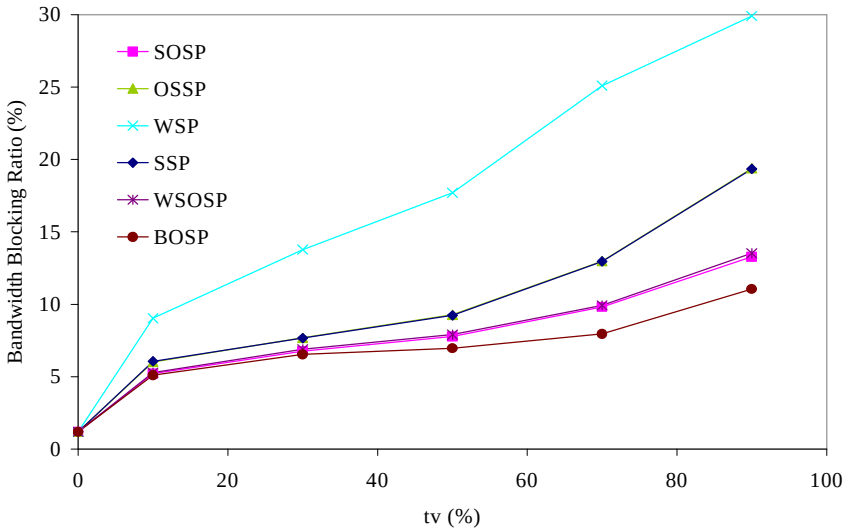


Figure 21. Bandwidth Blocking Ratio for Scenario 1 (Threshold triggering policy)

policies used in these simulations are Threshold and the Exponential class (with f = 2). The results presented have been obtained after repeating 300 seconds of simulation 10 times. The parameters used to measure the algorithms' behaviour are the routing inaccuracy and the bandwidth blocking ratio, that is, the same used to evaluate the basic *BBR* mechanism.

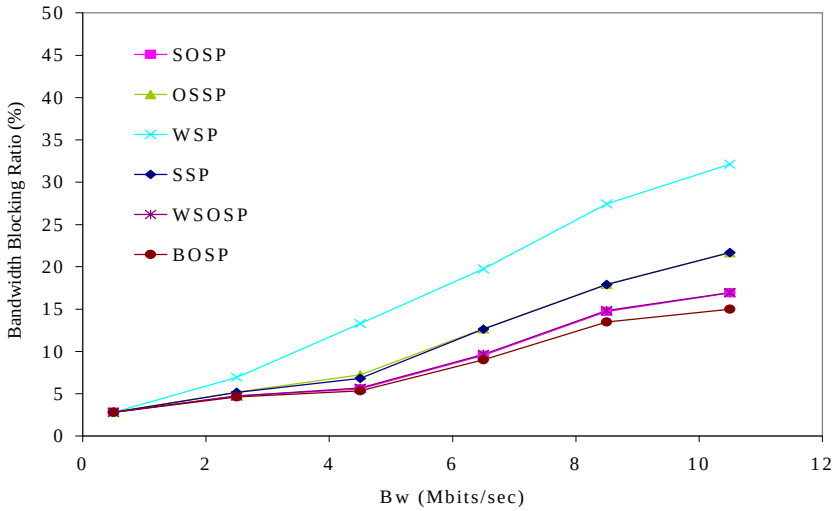


Figure 22. Bandwidth Blocking Ratio for Scenario 1 (Exponential class triggering policy)

Figure 21 and Figure 22 show the bandwidth blocking ratio tested for both the Threshold and Exponential class triggering policies respectively. Note that the best performance is obtained using *BOSP*, whereas *WSOSP* exhibits similar results to *SOSP*, and in both cases, *BOSP* and *WSOSP* present better results than *WSP*. In the worst conditions (the threshold *tv* of the triggering policy is 90%), the bandwidth blocking ratio obtained by *BOSP* (11%) substantially improves those obtained by *WSOSP* (13.5%), *SOSP* (13.3%) and *SSP* (19.3%). Recall that by increasing the threshold value the number of update messages flooded throughout the network is reduced.

Regarding routing inaccuracy behaviour, Figure 23 and Figure 24 show again that *BOSP* exhibits better results for both triggering policies. The number of paths incorrectly selected is extremely low even for large values of the threshold and the base class size.

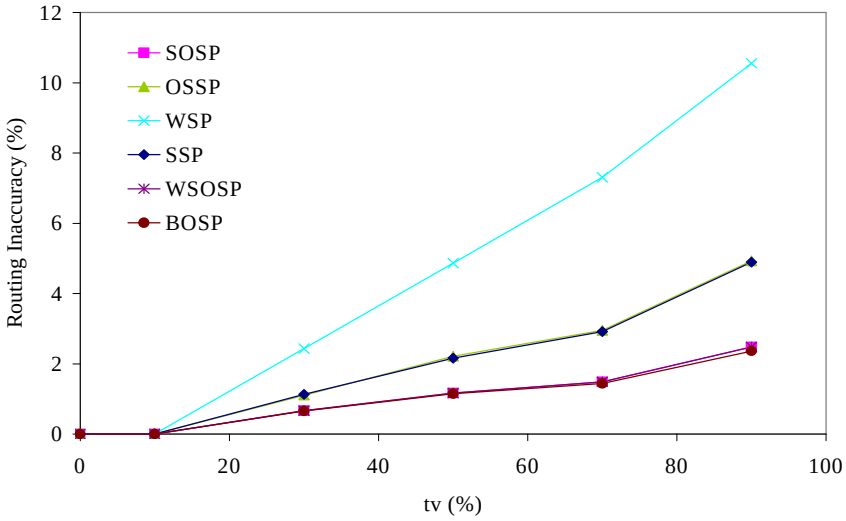


Figure 23. Routing Inaccuracy for Scenario 1 (Threshold triggering policy)

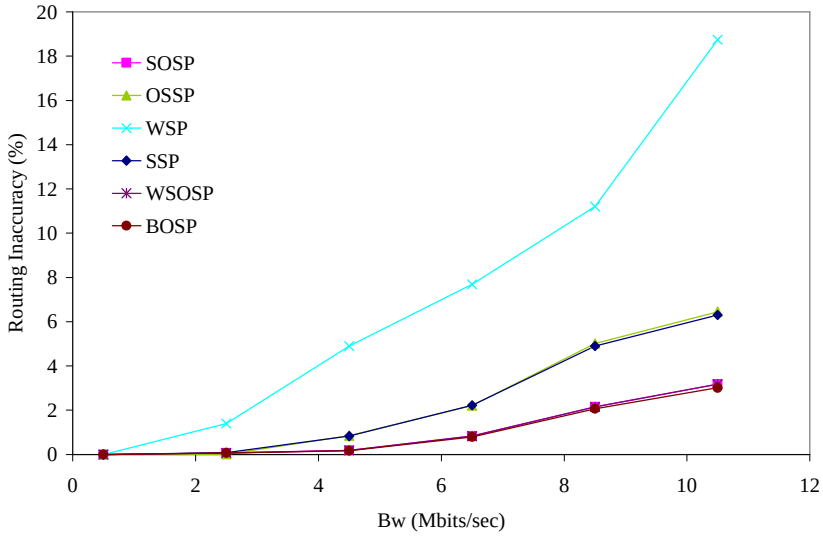


Figure 24. Routing Inaccuracy for Scenario 1 (Exponential class triggering policy)

Finally, Figure 25 and Figure 26 show the cost of using the *BBR* mechanism. As in previous Chapter the number of *bypass-paths* computed per route, *n_bp* is limited to 3. The number of computed *bypass-paths* grows when either the threshold or the base class size value increases. This is logical, since the number of *OSLs* grows when the amount of flooding messages decreases. It is important to observe that when *BOSP* is applied, not only do the bandwidth blocking ratio and the routing

inaccuracy decrease, but the cost decreases as well. This is due to the optimisation achieved in the path selection process.

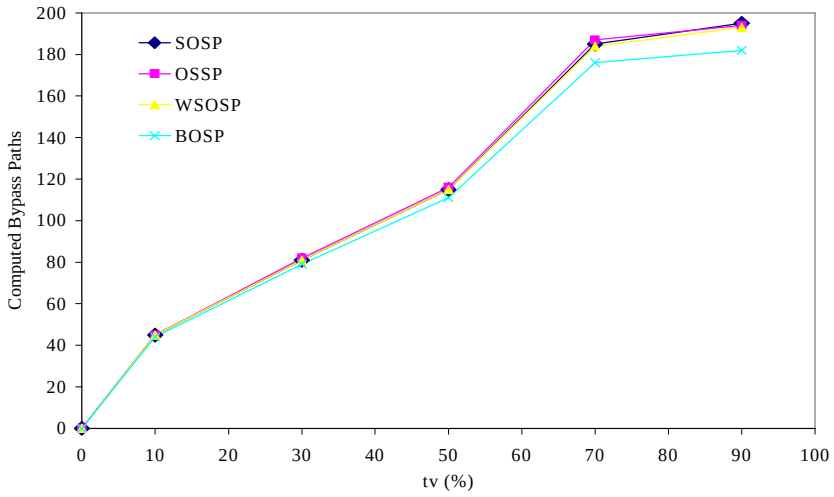


Figure 25. Computed *bypass-paths* for Scenario 1 (Threshold triggering policy)

Hence, the *BOSP* is the *BBR* algorithm which presents the best performance in terms of blocking probability, number of routes incorrectly selected and the cost because of computing *bypass-paths*.

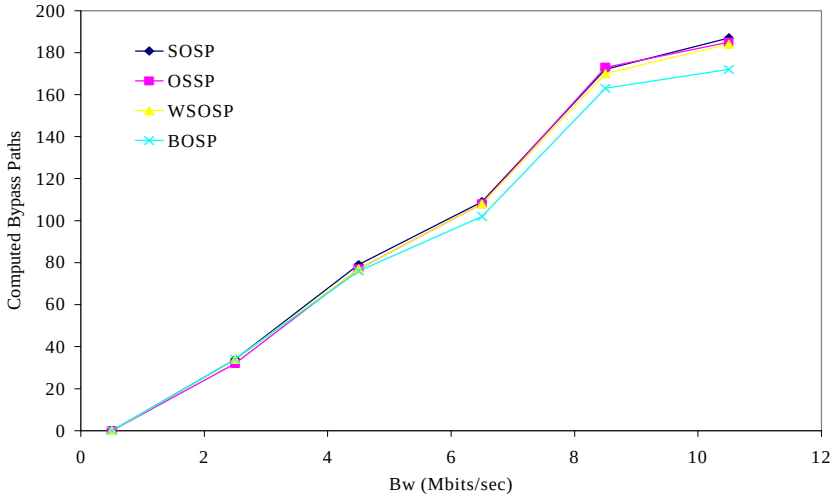


Figure 26. Computed *bypass-paths* for Scenario 1 (Exponential class triggering policy)

It is worth to notice that all simulations performed in Chapter 6 and Chapter 7 limit the number of computed *bypass-paths* per route to 3. As already explained, this is a restriction imposed to reduce the cost. It might be quite interesting to find out

which is the restriction imposed in the reduction obtained in the bandwidth blocking ratio because of limiting the number of *bypass-paths* computed per route. This evaluation is performed in two parts. On the one hand, the impact of limiting computed *bypass-paths* to a fixed value (1,2 and 3) on both bandwidth blocking probability and cost is analyzed by simulation in Chapter 8.

On the other hand, unlike the solution applied so far to reduce the cost, a different solution to reduce the cost of the *BBR* mechanism is to make the *n_bp* value dependent on the network load. When network is not highly loaded, a low number of links defined as *OSL* is expected, therefore a low number of *bypass-paths* per route might be computed. However, when network is heavily loaded a high number of *bypass-paths* per route are needed to cover *OSL* definition. Assuming that the larger the network load, i.e., the larger the number of *bypass-paths* computed, the lower the probability that *bypass-paths* might be really used, since they are also unavailable, the probability that a *bypass-path* may be used decreases as network load increases. Therefore another solution to reduce the cost of the *BBR* mechanism is to make the number of computed *bypass-paths* per route dependent on the network load. This approach is evaluated in the Scenario 2.

2) Scenario 2:

In this case, the *BBR* mechanism is tested over a realistic network topology to verify its influence on global network performance. The *ISP* topology shown in Figure 27 is a popular topology used in many old and recent QoS routing studies and is typical of the nationwide network of a US based ISP. There are three source nodes computing routes for any other destination node (except themselves).

Assuming the *SOSP* and the *BOSP* be those routing algorithms inferred from the *BBR* mechanism presenting lower blocking probability, the algorithms evaluated are the *WSP*, *SSP*, *SOSP* and *BOSP*. Unlike previous simulations where the number of computed *bypass-paths* per route is limited by fixing the *n_bp* value, in these simulations the cost is reduced by making the *n_bp* value dependent on the network load in the range from 0 *bypass-paths* per route (network highly loaded) to 5 *bypass-paths* per route (network not loaded). Once more, the bandwidth blocking ratio and the routing inaccuracy are the parameters evaluated. In previous simulations, both

parameters present similar behaviour for both the Threshold and the Exponential class triggering policies. Assuming this as a constant behaviour, in the following simulations only the Threshold triggering policy is analyzed.

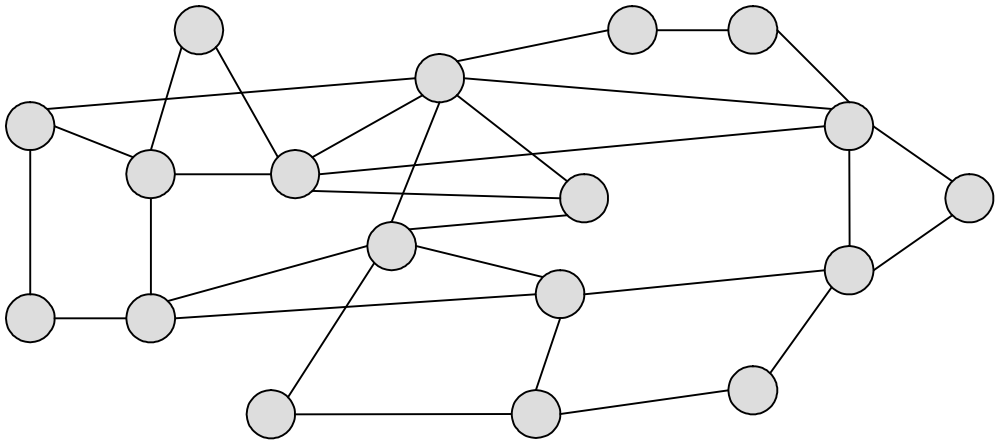


Figure 27. The *ISP* topology used in simulations

All links are assumed to be bi-directional and of the same capacity of 622 Mb/s. Every simulation requests 2500 *LSP*s which arrive following a Poisson distribution, where the requested bandwidth is uniformly distributed between 1Mb/s and 5Mb/s. The holding time is randomly distributed with a mean of 120 sec. Simulations run during 250 sec.

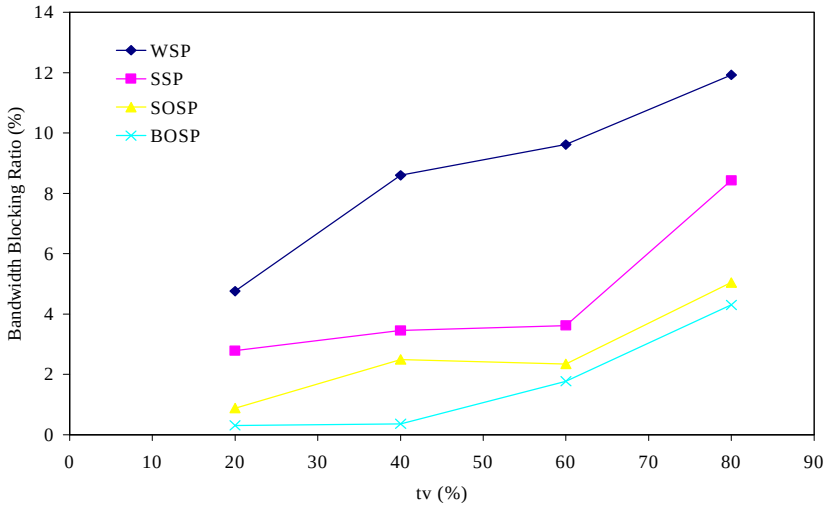


Figure 28. Bandwidth Blocking Ratio for Scenario 2 (Threshold triggering policy)

Figure 28 exhibits the bandwidth blocking ratio obtained by simulation. Again, while the *SOSP* behaves better than the *SSP*, the *BOSP* provides the lower blocking probability. It is worth to notice that the larger difference in the obtained blocking reduction when using the *BOSP* algorithm compared to the *WSP* and *SSP* algorithms is obtained for *tv* = 60% instead of *tv* = 80%.

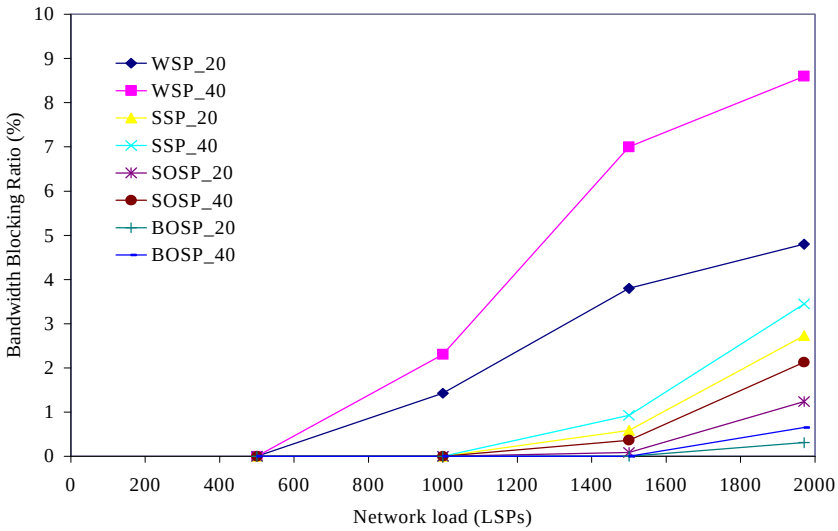


Figure 29. Bandwidth Blocking Ratio and network load for Scenario 2 (*tv* = 20%, *tv* = 40%)

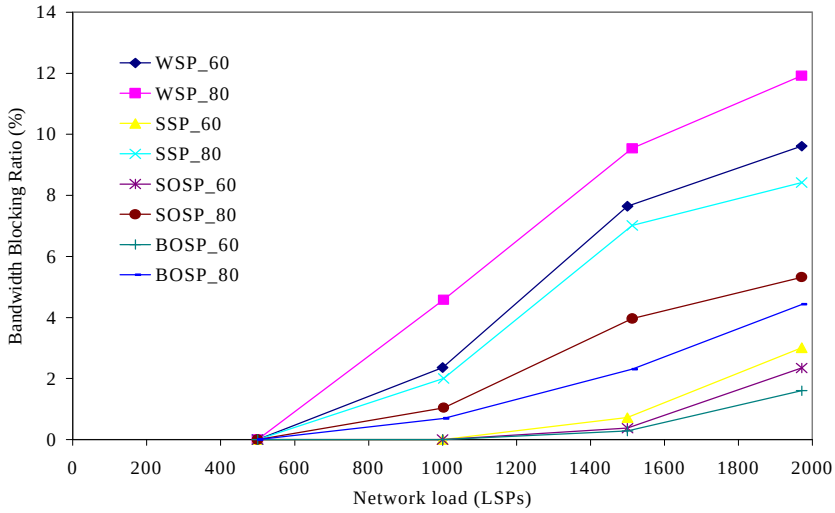


Figure 30. Bandwidth Blocking Ratio and network load for Scenario 2 (*tv* = 60%, *tv* = 80%)

The evolution of the bandwidth blocking as a function of network load in terms of established *LSP* connections is analysed in next Figures. Meanwhile Figure 29

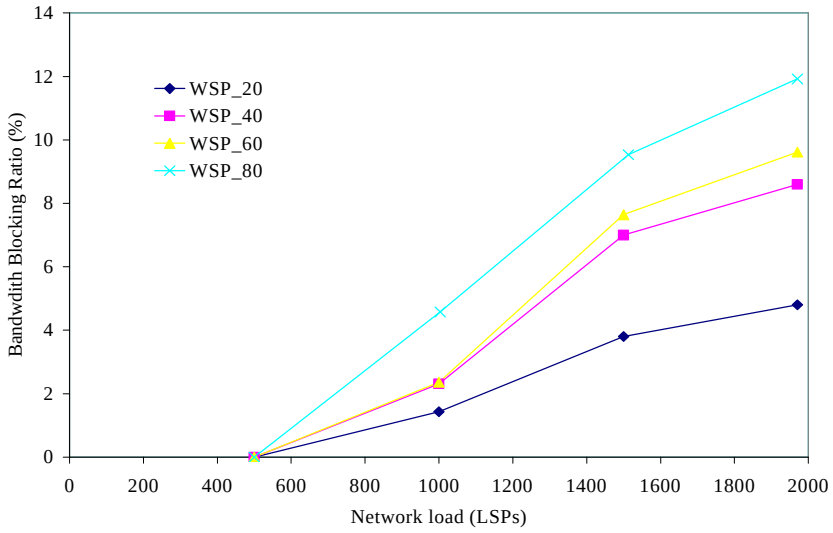


Figure 31. *WSP* algorithm behaviour as a function of the network load (Scenario 2)

exhibits the blocking behaviour for *tv* = 20% and *tv* = 40%, Figure 30 represents the blocking behaviour for *tv* = 60% and *tv* = 80%. Both Figures show the benefits in terms of bandwidth blocking reduction obtained when applying algorithms inferred from the *BBR* mechanism for any *tv* value. It is worth to notice that for low network load and medium values of *tv* the *SOSP* performs almost better than the *BOSP* algorithm. Therefore, the *BOSP* algorithm gives larger benefits on bandwidth blocking reduction for high network loads and medium values of *tv*.

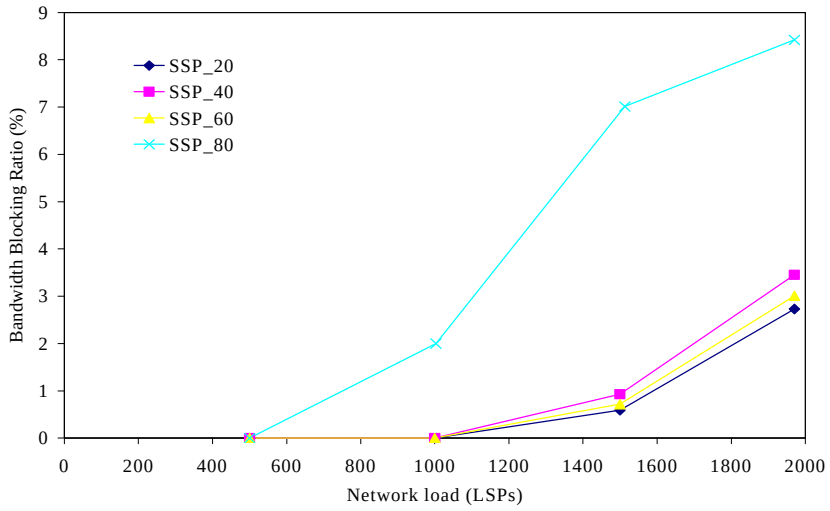


Figure 32. *SSP* algorithm behaviour as a function of the network load (Scenario 2)

Next figures draw the behaviour of each algorithm as a function of the network load in terms of number of established *LSP*s for each threshold value. In this way it is possible to independently observe the evolution in the bandwidth blocking depending on the *tv* value on each algorithm. The *WSP*, *SSP*, *SOSP* and *BOSP* are shown in Figure 31, Figure 32, Figure 33 and Figure 34 respectively.

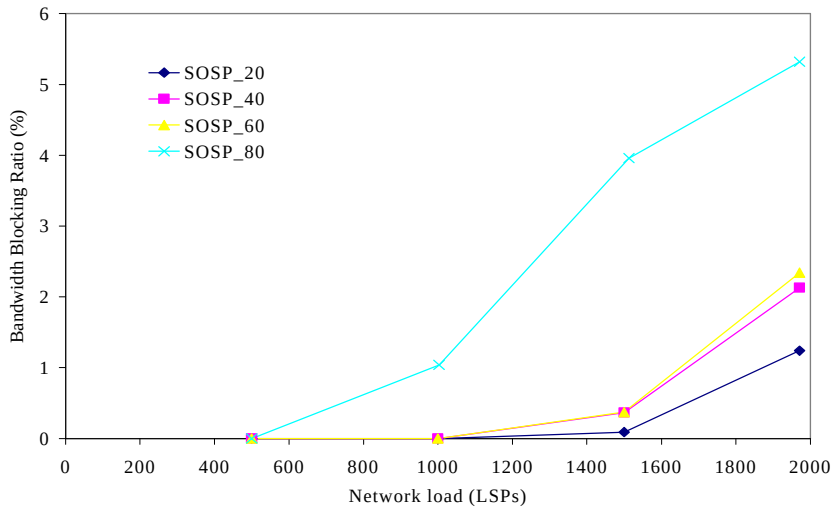


Figure 33. *SOSP* algorithm behaviour as a function of the network load (Scenario 2)

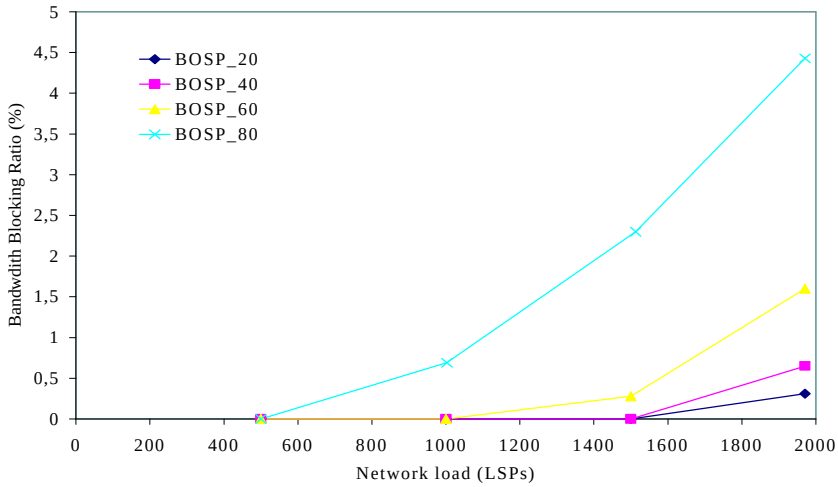


Figure 34. *BOSP* algorithm behaviour as a function of the network load (Scenario 2)

The number of routes incorrectly selected is shown in Figure 35. Again, the *BOSP* is the algorithm that computes the lower number of incorrect routes. As previous

simulations, *BOSP* and *SOSP* algorithms present better behaviour compared to the *SSP* and the *WSP*.

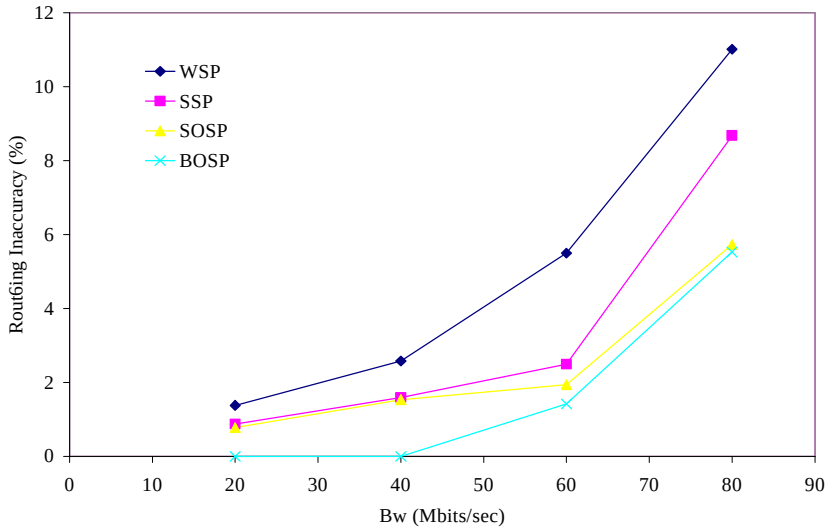


Figure 35. Routing Inaccuracy for Scenario 2 (Threshold triggering policy)

Figure 36 draws the cost of using the *BBR* mechanism. Remind that in these simulations *n_bp* is not a constant value instead it is network load dependent. The total number of computed *bypass-paths* is substantially reduced in comparison with values obtained in previous simulations. However, someone could argue that the reduction obtained in the cost might negatively reduce the *BBR* introduced benefits.

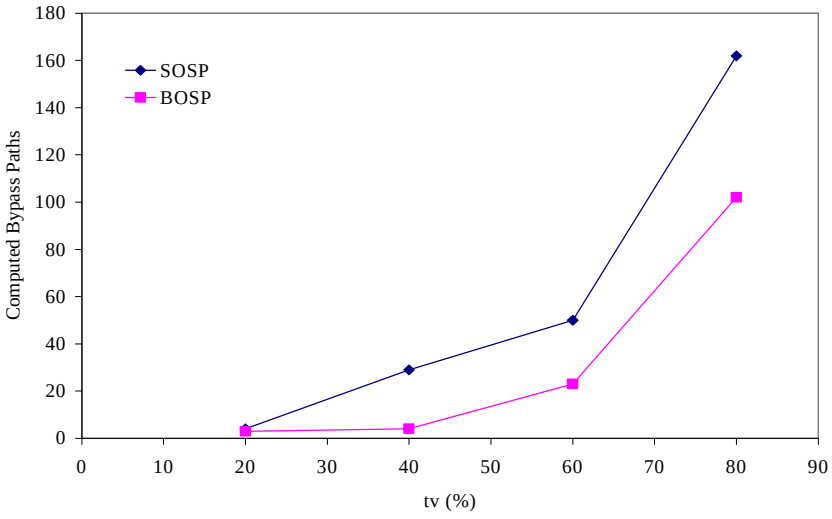


Figure 36. Computed *bypass-paths* for Scenario 2 (Threshold triggering policy)

Table 6 compares the reduction in the bandwidth blocking obtained by the *BOSP* under three different situations: when *n_bp* = 3, when *n_bp* is not limited (a ∞ number of *bypass-paths* might be computed per route) and when *n_bp* is network load dependent.

Table 6. Cost analysis

n_bp	BW_{WSP}	BW_{SSP}	*Cost (bypass-paths)*	*Cost (time)*
3	8.24	4.62	282	14.30%
Not limited	15.11	7.99	3178	161.23%
Network load dependent	7.62	4.13	102	5.71%

The values BW_{WSP} and BW_{SSP} stand for the difference in the bandwidth blocking obtained when comparing the *BOSP* to the *WSP* and the *SSP* respectively. The cost is represented in terms of both the total number of computed *bypass-paths* and the impact of computing these *bypass-paths* on the computational time. The second situation, when *n_bp* is not limited, is really not affordable because of the extreme cost. Analysing the first and the last situation, while similar results in bandwidth blocking are obtained the cost is significantly reduced when distributing *n_bp* proportionally to the network load.

3) Scenario 3:

This simulation is performed to analyse the impact on the *BBR* performance as a function of the link capacity. Simulations are performed over the network topology shown in Figure 27 although some variations are considered. In this case all links are assumed to be bi-directional with a capacity of 2.5 Gb/s. Every simulation requests 10500 *LSP*s which arrive following a Poisson distribution, where the requested bandwidth is uniformly distributed between 1Mb/s and 5Mb/s. The holding time is randomly distributed with a mean of 120 sec. A number of 10 simulations have been carried out each one lasting 295 sec. Again, the same parameters are evaluated and the same algorithms are implemented, *WSP*, *SSP*, *SOSP* and *BOSP*.

There are two conclusions that can be extracted after analysing the bandwidth blocking shown in Figure 37. First, while all the algorithms suffer from an increment in the bandwidth blocking, the *BOSP* algorithm is still maintaining a quasi linear behaviour. Second, there is a large difference in the bandwidth blocking obtained by

the *BOSP* algorithm in comparison with the other algorithms. In fact, while in previous simulation the bandwidth blocking difference obtained by the *BOSP* compared to the *SSP* is 3.99 %, in this simulation this difference is 5.7%.

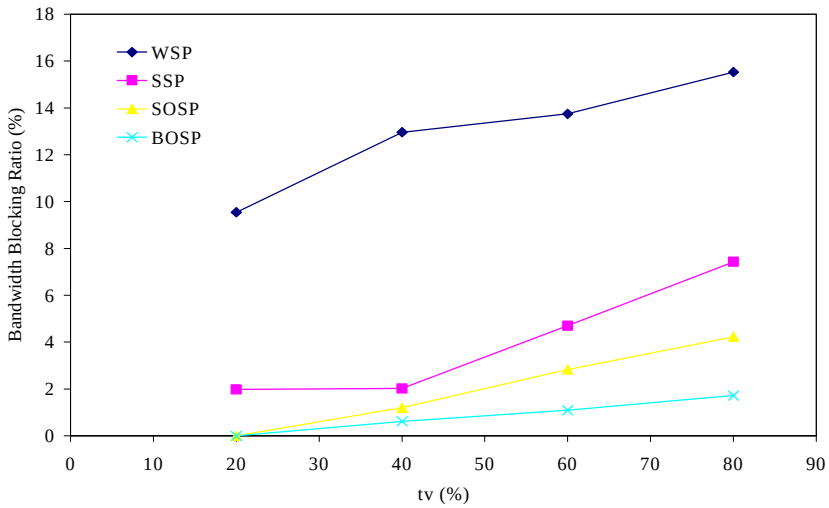


Figure 37. Bandwidth Blocking Ratio for Scenario 3 (Threshold triggering policy)

Figure 38 shows the impact on the blocking as a function of the network load in terms of number of established *LSP*s for values of *tv*= 20% and 40%. In Figure 39 *tv* values are 60 % and 80%.

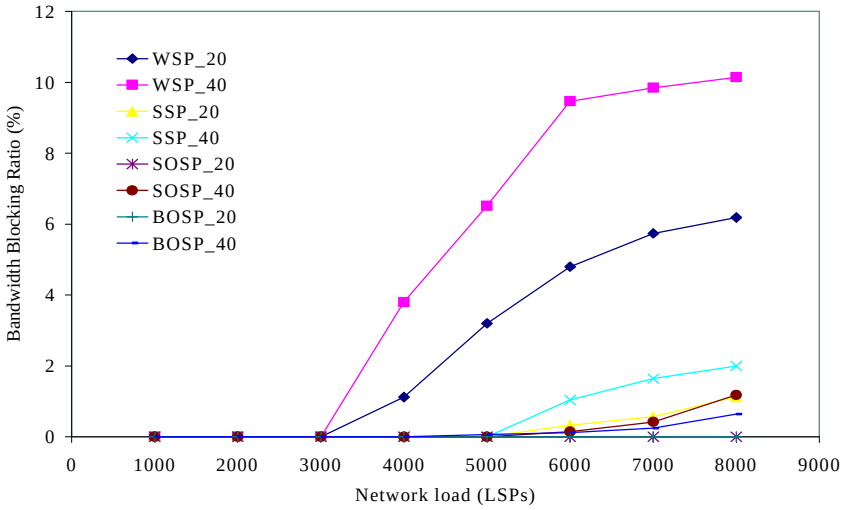


Figure 38. Bandwidth Blocking Ratio and network load for Scenario 3 (*tv* = 20 %, *tv* = 40%)

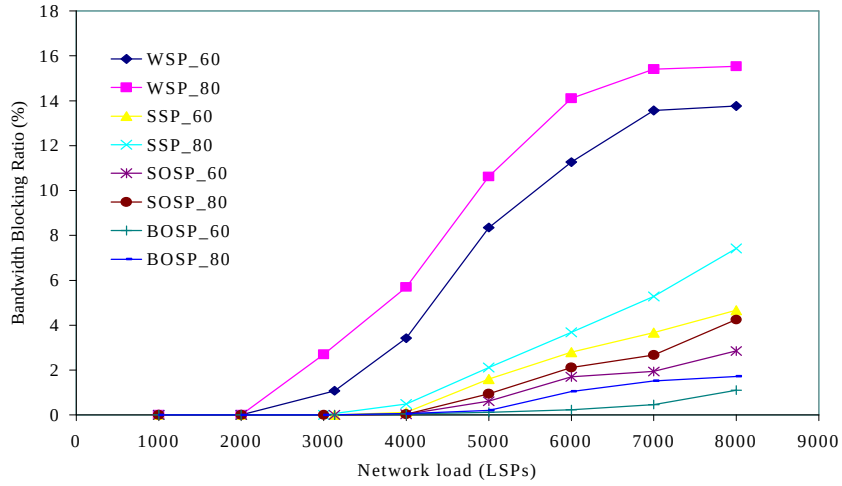


Figure 39. Bandwidth Blocking Ratio and network load for Scenario 3 (*tv* = 60%, *tv* = 80%)

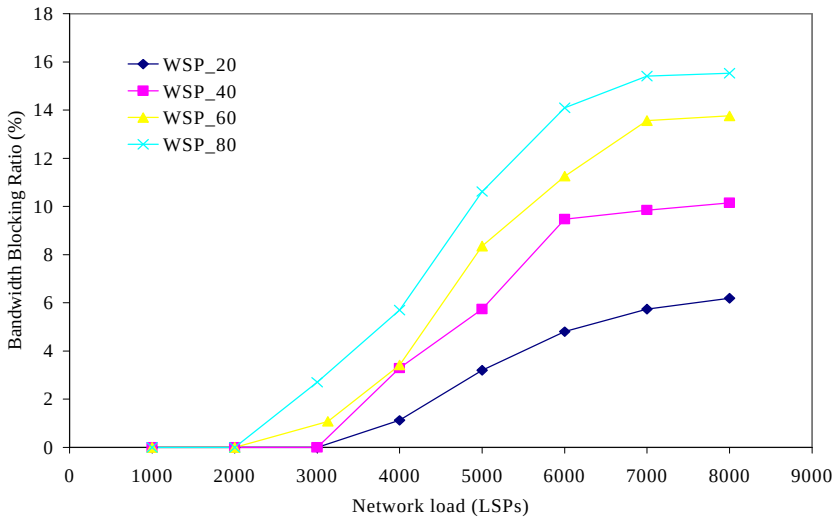


Figure 40. *WSP* algorithm behaviour as a function of the network load (Scenario 3)

Again, next figures draw the behaviour of each algorithm as a function of the network load in terms of number of established *LSP*s for each threshold value. Figure 40, Figure 41, Figure 42 and Figure 43 show the bandwidth blocking evolution for the *WSP*, *SSP*, *SOSP* and *BOSP* algorithms respectively.

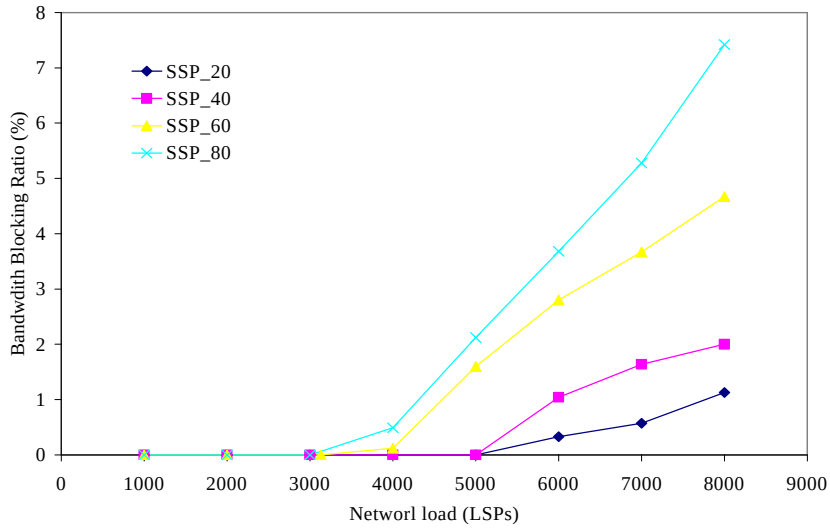


Figure 41. *SSP* algorithm behaviour as a function of the network load (Scenario 3)

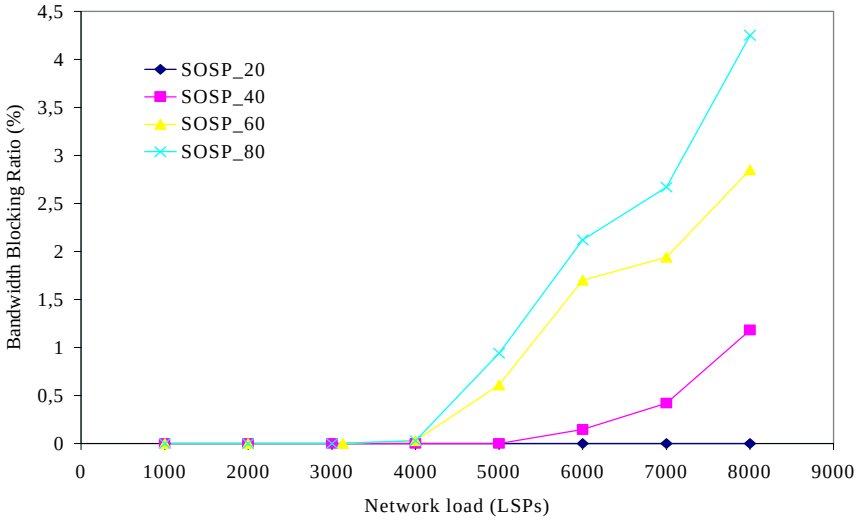


Figure 42. *SOSP* algorithm behaviour as a function of the network load (Scenario 3)

Figure 44 shows the routing inaccuracy for all the evaluated algorithms when applying the Threshold triggering policy. The *SOSP* and the *BOSP* algorithms exhibit a very similar behaviour.

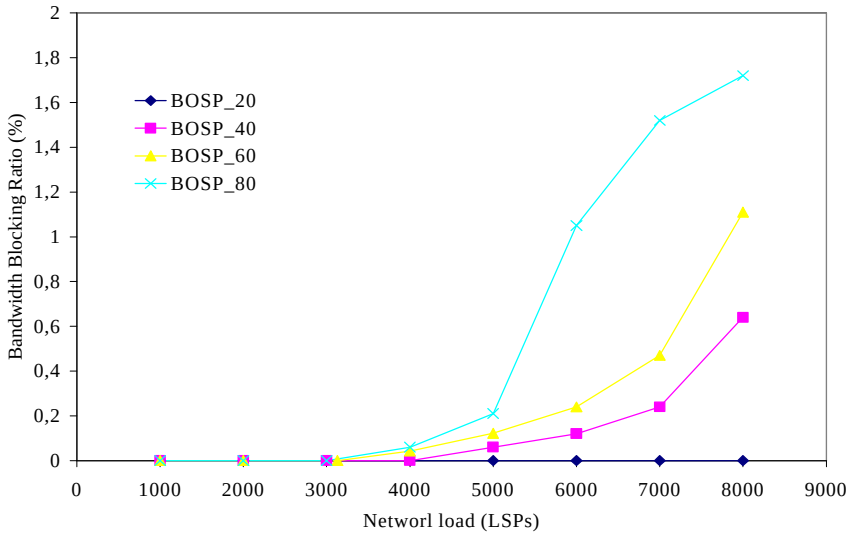


Figure 43. *BOSP* algorithm behaviour as a function of the network load (Scenario 3)

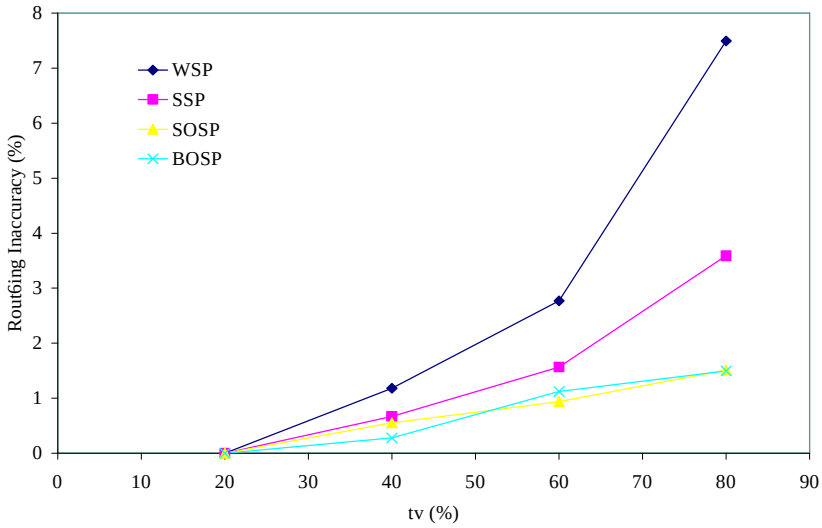


Figure 44. Routing Inaccuracy for Scenario 3 (Threshold triggering policy)

Finally, Figure 45 plots the cost in terms of computed *bypass-paths*. Only a negligible increment in the number of computed *bypass-paths* is produced. Therefore, while *BBR* mechanism is regardless of links size in terms of the number of routes incorrectly selected, the obtained bandwidth blocking ratio suffers from lower reduction.

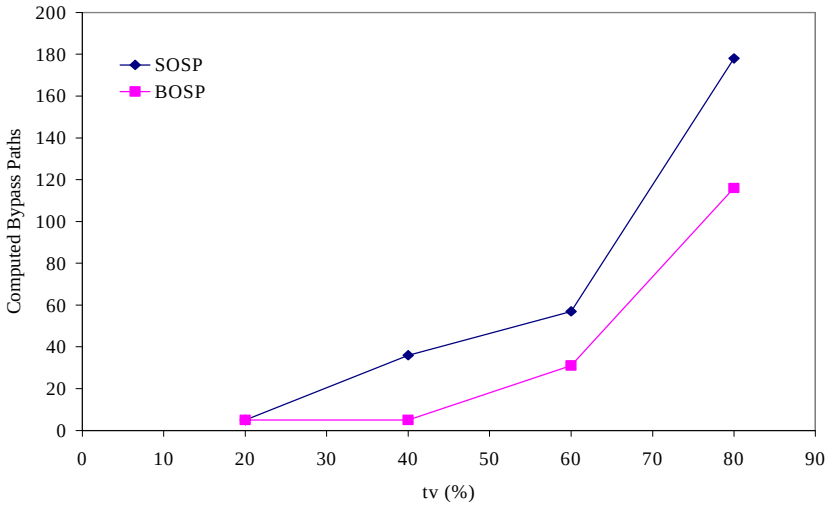


Figure 45. Computed *bypass-paths* for Scenario 3 (Threshold triggering policy)

Summarizing, in this Chapter it has been extensively proved that the larger the number of computed *bypass-paths* the lower the blocking probability reduction, but also the larger the cost. It has also been shown that when the *n_bp* value is dependent on the network load the cost of applying the *BBR* mechanism, specifically the *BOSP* algorithm, is substantially reduced while variation on the obtained bandwidth blocking is hardly significant.

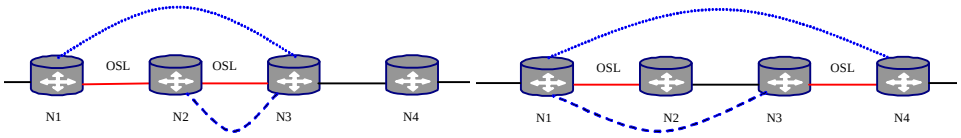
The next Chapter presents the *BYPASS Discovery Process*. Moreover, it is also analyzed the evolution in the blocking ratio reduction as a function of the number of computed *bypass-paths* per route.

Chapter 8

BYPASS Discovery Process

The *BYPASS Discovery Process (BDP)* appears as a solution to extend the *BBR* applicability. In fact, as already stated in previous Chapters, there are some network scenarios where the *BBR* mechanism cannot be applied since a *bypass-path* cannot be found. This occurs because the *bypass-path* definition requires that the edge nodes of the *bypass-path* to be computed must perfectly match the edge nodes of the *OSL* that bypasses. Therefore, whenever an alternative and disjoint route between the edges nodes of that link defined as an *OSL* cannot be found, the *BBR* mechanism cannot be applied. In this case, instead of rerouting the set-up message along the *bypass-path* that should have been computed the set-up message will be blocked and dropped when the selected route lacks enough bandwidth.

The *BDP* addresses this problem adding a modification in the *BBR* mechanism which extends the mechanism used to select *bypass-paths* guaranteeing a higher number of computed *bypass-paths*. Let i_j and e_j be the edge nodes of a link $l^{os}_j \hat{I} L^{os}$.



(a) (b)

Figure 46. *BDP* process

Let e_{j+k} be the *k* node adjacent to e_j downstream along the working path. Then, *BDP* computes *bypass-paths* in accordance with the following rules:

- Look for an alternative and disjoint route between the (i_j,e_j) pair (as done in the normal *BBR* mechanism).
- If there is not a feasible disjoint route between the (i_j,e_j) pair, then look for a route between the (i_j,e_{j+k}) pair, for $1 \leq k \leq d$, being e_{j+d} the destination node.

Three main aspects must be analyzed when applying the *BDP*. Firstly, feasible bypass routes cannot include any node belonging to the working path but the egress (destination) node. Secondly, Rule 2 in the *BBR* definition is still meaningful. Therefore, when two adjacent nodes are defined as *OSL*s the *BDP* computes a *bypass-path* to bypass both links and then another *bypass-path* to bypass only the second link, as shown in Figure 46 (a). Finally, consider the scenario shown in Figure 46 (b) where links N1-N2 and N3-N4 are defined as an *OSL* (if link N2-N3 is also defined as an *OSL* this case matches Rule 2). Suppose that the dash line stands for a possible *bypass-path* to bypass link N1-N2. In this situation, to reduce the possible number of *bypass-paths* used, an optimal *bypass-path* would be that drawn by the dot line from N1 to N4. Then, a *bypass-path* must also be computed to bypass link N3-N4 (not drawn in Figure 46 (b)).

8.1 Example to Illustrate the BDP Performance

Figure 47 is used to ease the *BDP* understanding. The performance of the *SOSP* and the *BOSP* (the two *BBR* algorithms better performing) algorithms when both include the *BDP* are analyzed. Suppose that update messages are sent according to the Exponential Class triggering policy with f=2 and Bw=1. Assuming that an

incoming *LSP* request arrives at LSR0 demanding b_{req} of 4 units of bandwidth between LSR0 to LSR4, Table 7 shows different routes from LSR0 to LSR4 including the parameters used by the *SOSP* and the *BOSP* algorithms to select the path.

Table 7. *BBR* Process when including the *BDP*

Id	*Route (LSR)*	*H*	*OSL*	b_r^{min}	F_p
a	0-1-2-3-4	4	1	4	1
b	0-1-5-6-7-4	5	2	7	0.71
c	0-1-5-2-3-4	5	1	6	0.83
d	0-8-9-4	3	3	4	0.75

BBR	*SOSP*	*BOSP*
1th step	Mark OSLs	Mark OSLs
2on step	a,c	a,c
3th step	a	c
4th step	1-2 (1,5,2)	5-2 (5,6,7,4)

It is also shown in detail in Table 7 the four steps realized by the *BBR* mechanism to select paths. Specifically the *SOSP* and the *BOSP* algorithms are depicted. The 4th step, *bypass-path* selection, includes the *BDP* mechanism proposed in this Chapter.

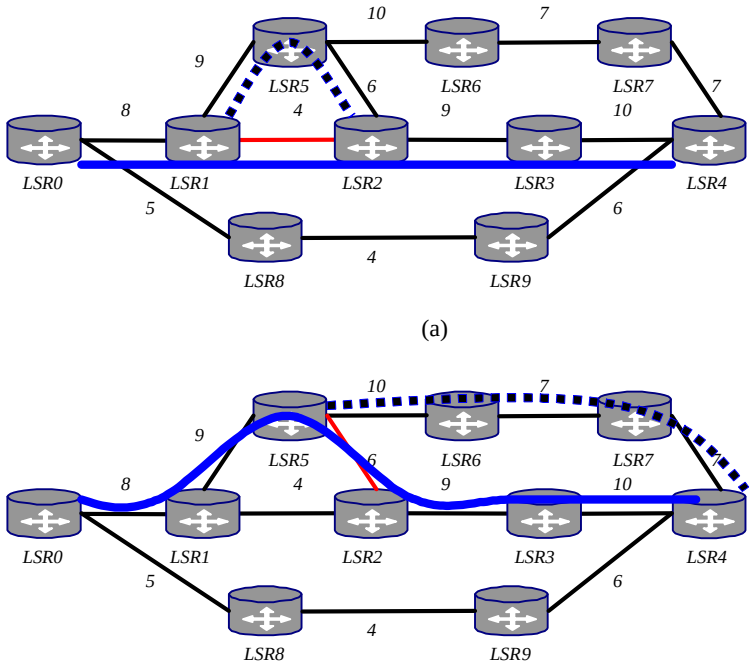


Figure 47. *BDP* performance: illustrative example

According to the *SOSP* behaviour, Figure 47 (a), a *bypass-path* exists (LSR1, LSR5, LSR2) to bypass the edges nodes of the link defined as an *OSL* that is LSR1-LSR2, represented by a dash line. However, when applying the *BOSP* algorithm, Figure 47 (b), there is not a *bypass-path* that directly bypasses the edge nodes of the

link defined as *OSL*, namely LSR5-LSR2. In this case, if the *BDP* mechanism is not implemented, a *bypass-path* might not be computed. *BDP* allows the *BBR* mechanism to compute a *bypass-path* (to bypass the *OSL* between nodes LSR5 and LSR2) from LSR5 to LSR4 (destination node), represented by a dash line. This *bypass-path* will be used if there are insufficient bandwidth in the link defined as an *OSL* when the *Path* message reaches LSR5, hence improving the *BBR* applicability and so reducing the bandwidth blocking ratio.

8.2 Performance Evaluation

In this Section we compare by simulation the benefits introduced when including the *BDP* in the *BBR* mechanism. Being the *BOSP* the best algorithm inferred from the *BBR* mechanism, in terms of blocking ratio and optimal path selection, the algorithms evaluated are the *Shortest-Safest Path* (*SSP*), and the *BOSP*. In order to clearly identify the improvement obtained by the *BBR* mechanism when the *BDP* is also applied, the *BOSP* algorithm is evaluated in both situations, i.e., when it does not include the *BDP* (named *BOSP*) and when includes the *BDP* (named *B/BDP*). Moreover, in order to evaluate the impact on the blocking probability because of the number of *bypass-paths* that can be computed per route, different simulations are carried out as a function of *n_bp*. Remind that *n_bp* has been defined as a bundle showing the maximum number of *bypass-paths* that can be computed on each working path. The notation used in the figures to denote the *n_bp* values is *BOSP(n_bp)* and *B/BDP(n_bp).* The simulations are performed over the network topology shown in Figure 11, using the ns/2 simulator extended with *MPLS, BBR* and *BDP* features. We use two link capacities, 622 Mb/s represented by a light line and 2.5 Gb/s represented by a dark line. Every simulation requests 2500 *LSP*s which arrive following a Poisson distribution where the requested bandwidth is uniformly distributed between 1 Mb/s and 5 Mb/s. The holding time is randomly distributed with a mean of 120 seconds. The Threshold and the Exponential class (with $f = 2$) triggering policies are evaluated. Results have been obtained after repeating 300 seconds of simulation 10 times. As previous simulations, the parameters used to measure the algorithms behaviour are the routing inaccuracy and the bandwidth

blocking ratio. The bandwidth blocking ratio for the Threshold and Exponential class triggering policies are depicted in Figure 48 and Figure 49 respectively.

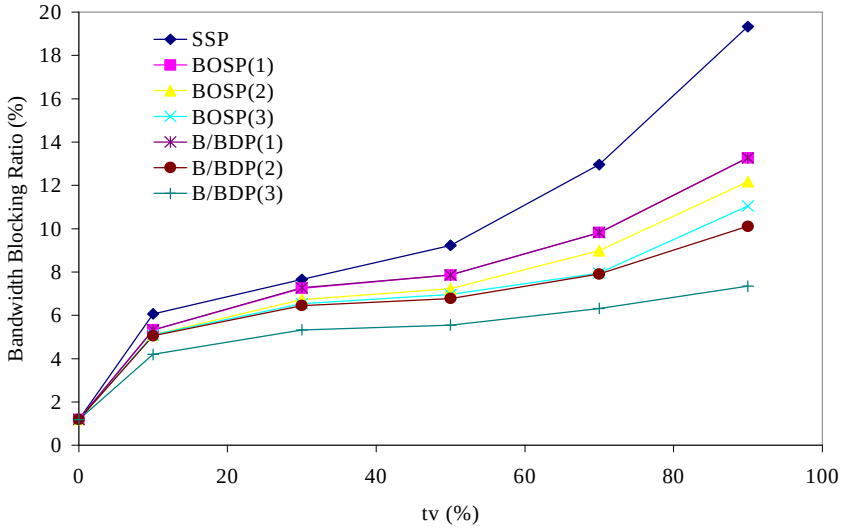


Figure 48. Bandwidth Blocking Ratio for the Threshold triggering policy

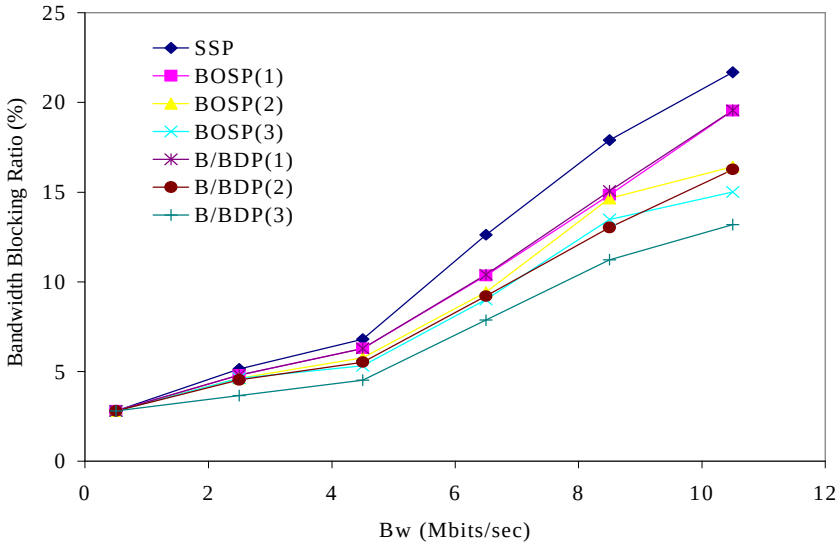


Figure 49. Bandwidth Blocking Ratio for the Exponential class triggering policy

Focusing, for instance, on the Threshold triggering policy a lower blocking is obtained when the *BBR* mechanism is applied compared to the *SSP* algorithm. Several conclusions can be obtained when analyzing Figure 48 in detail. Firstly, the lowest blocking is obtained when the *BDP* is applied, i.e., by the *B/BDP* algorithm. Secondly, although applying the *BDP* when *n_bp* = 1 hardly affects on the blocking

ratio, we can conclude that the effects produced when including the *BDP* in the *BBR* mechanism are completely dependent on the *n_bp* value. In fact, whereas a similar blocking is obtained for the *BOSP(1)* and the *B/BDP(1)* algorithms, (13.27 % and 13.3 % respectively) a significant reduction of 2 % is obtained when *n_bp* = 2. Increasing the *n_bp* values leads to obtain an even more significant blocking reduction. Hence, the larger the number of computed *bypass-paths* per route the lower the blocking, that is, the blocking depends on the computational cost introduced in the network. Lastly, the larger the threshold value, i.e., the inaccuracy, the larger the improvement in the blocking reduction. In the worst conditions (the threshold *tv* of the triggering policy is 90 %), the bandwidth blocking ratio obtained when applying the *BDP* process substantially improves that obtained by only applying the *BOSP*. For instance, when *n_bp* = 3, the obtained reduction in the blocking ratio is about 3.75 %.

Figure 50 and Figure 51 depict the number of routes incorrectly selected for the Threshold and the Exponential class triggering policies respectively. A similar behaviour is obtained for both triggering policies. As expected, the number of routes incorrectly selected decreases with the number of computed *bypass-paths*. Focusing again on the Threshold triggering policy, in the worst conditions (*tv* = 90 %), a reduction of 1.3 % is obtained when applying the *B/BDP(3)* algorithm compared to the *BOSP(1)* algorithm.

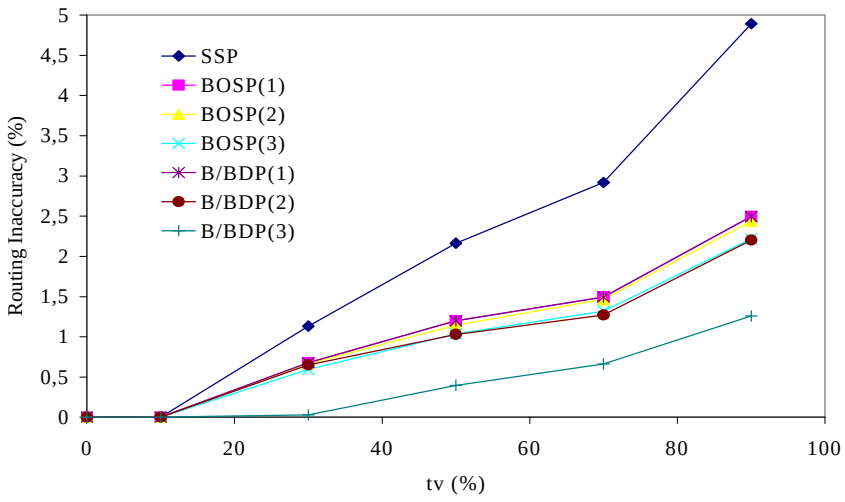


Figure 50. Routing Inaccuracy for the Threshold triggering policy

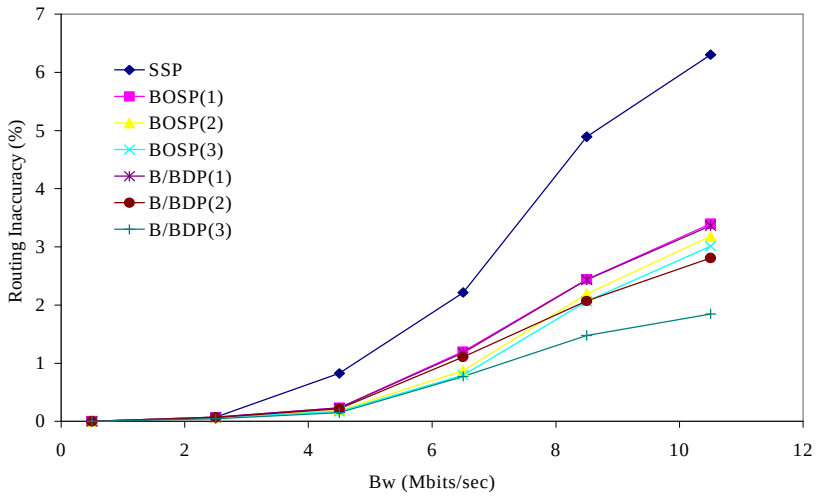


Figure 51. Routing Inaccuracy for the Exponential class triggering policy

Finally, Figure 52 and Figure 53 show the cost of applying the *BBR* mechanism with and without *BDP* capabilities. Moreover, it is also drawn the variation in the cost produced as a function of the *n_bp* value.

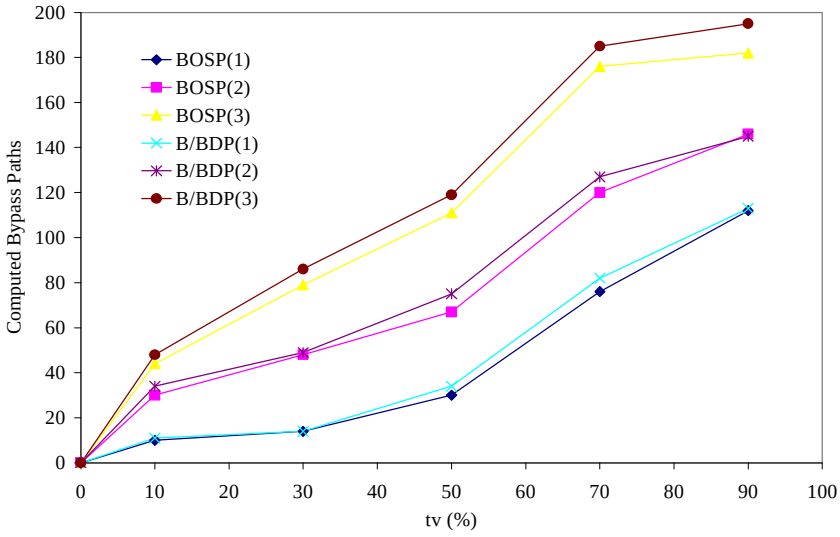


Figure 52. Computed *bypass-paths* for the Threshold triggering policy

Definitely, we can conclude that including the *BDP* in the *BBR* mechanism substantially improves global network performance.

In this Chapter the obtained bandwidth blocking reduction has been evaluated depending on two factors. On the one hand, it has been shown that introducing the

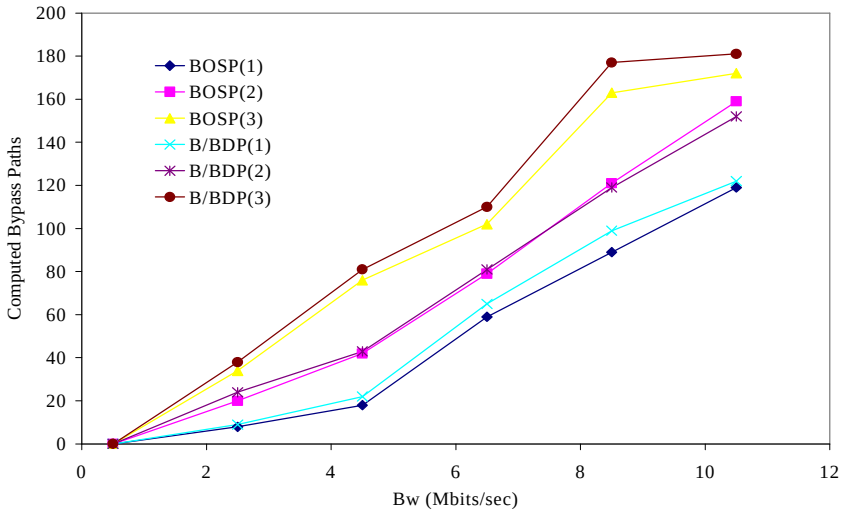


Figure 53. Computed *bypass-paths* for the Exponential class triggering policy

BDP process in the *bypass-path* computation significantly reduces the bandwidth blocking. This is because a high number of *bypass-paths* may be computed to bypass links defined as an *OSL*. On the other hand, different values of the *n_bp* parameter have been considered. Simulations have been performed for *n_bp* values of 1, 2 and 3. This means that only 1, 2 or 3 *bypass-paths* per route can be computed and used respectively. As expected, the larger the *n_bp* value the larger the blocking reduction, the larger the cost (in terms of total number of computed *bypass-paths*) and the lower the signalling overhead, i.e., number of update messages flooded throughout the network.

Summarising, the *BBR* mechanism is proposed to address the routing inaccuracy problem also introduced in this Part. Some routing algorithms are inferred from the *BBR* mechanism when considering either hopcount or load balance. *BBR* performance is enhanced both by adding the *BDP* process and by fixing the computational cost according to the network load. Results obtained by simulation show the benefits on global network performance when using the *BBR* mechanism.

PART III

ROUTING IN WDM NETWORKS

After introducing the need of optical networks for supporting current and future network models, in this Part the problem to be addressed is clearly presented and justified. The problem is an extension of the routing inaccuracy problem already tackled in an *IP/MPLS* scenario in previous Part of the Thesis. Therefore, this Part serves to extend the *BYPASS Based Routing* (*BBR*) solution proposed to address the routing inaccuracy problem in an *IP/MPLS* scenario so that it can be applied to *WDM* networks. In successive Chapters the proposed solution, called *BYPASS Based Optical Routing* (*BBOR*) is defined, illustrated in some network examples and finally evaluated by simulation. Depending on the wavelength-continuity constraint, different wavelength assignment algorithms are proposed, compared and evaluated.

Chapter 9

Routing and Wavelength Assignment in WDM Networks

In previous Chapters, the *BBR* mechanism has been applied to *IP/MPLS* networks to address the effects of inaccurate network state information on global network performance. In this Chapter a new mechanism extended from the *BBR* mechanism is also proposed. This mechanism, *BYPASS Based Optical Routing* (*BBOR*), copes with the routing inaccuracy problem in an optical scenario, reducing the negative effects of selecting lightpaths under inaccurate network state information. The *BBOR* mechanism is applied to networks without conversion capabilities [67] and to networks with conversion capabilities [68].

9.1 Introduction

In recent years the introduction of high capacity and reliable transport networks has become necessary in order to cover Internet traffic demands. New Internet applications increasingly request greater capacity and guarantees of traffic delivery

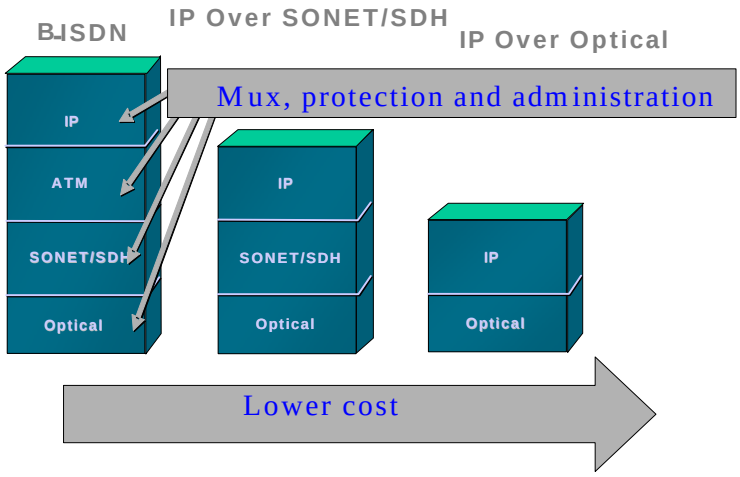


Figure 54. Network evolution

in such a way that the traffic transmission model must be modified. In fact, the network model is evolving to an *Optical Transport Network* (OTN) as shown in Figure 54.

An *OTN* consists of switching nodes (*Optical Cross-Connect*, *OXC*) interconnected by *wavelength-division multiplexed* (*WDM*) fibre-optic links that provide multiple huge bandwidth communication channels over the same fibre in parallel. A wavelength routed *WDM* network is a circuit-switched network, in which a lightpath must be established between a source-destination pair before data can be transferred. A lightpath is a unidirectional end-to-end connection between a source-destination pair, which may span multiple fibre links and use a single or multiple wavelengths. When the *OTN* includes automatic switching capabilities, it is referred to as an *Automatic Switched Optical Network* (*ASON*). Figure 55 depicts the ASON architecture. ASON must include a Control Plane, necessary to provide the network with dynamic provisioning, fast protection, restoration and *Traffic Engineering*. The *IETF* proposed *Generalized Multiprotocol Label Switching* (*GMPLS*) as a protocol to implement this Control Plane. In [69] a different solution to implement the Control Plane is discussed.

This Control Plane includes a lightpath control mechanism to efficiently set up and tear down lightpaths, which may be either centralized or distributed. In the former case, a single central controller having complete global network state

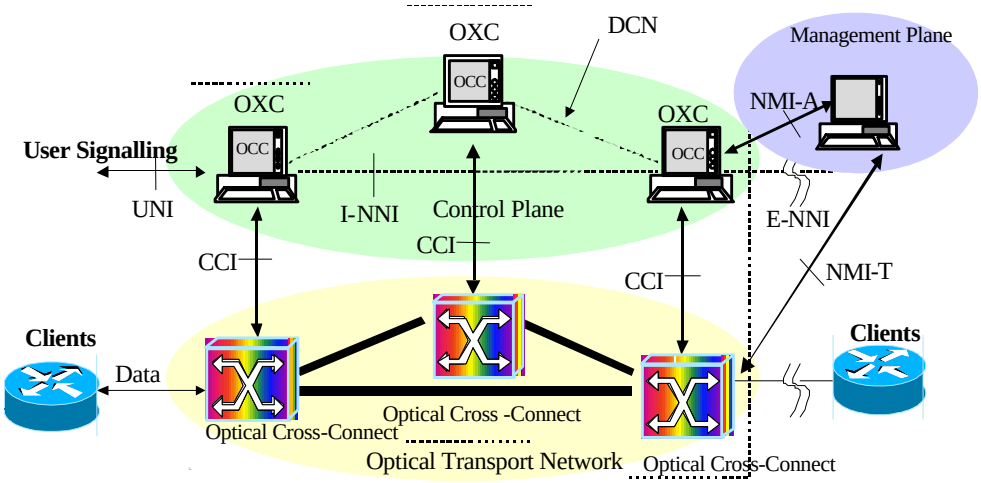


Figure 55. ASON architecture

information sequentially selects and establishes a lightpath for any incoming request. In the latter case, all incoming connection requests are simultaneously processed at different network nodes, which select the lightpaths based on either local (the nodes have not information about the whole network) or global network state information. On the one hand, if the routing decision is taken based on local information the probability that the set-up message will be rejected in any intermediate node is very large. On the other hand, using global network state information reduces the blocking probability, whenever this information represents a current picture of the network state. In spite of the fact that adaptive routing mechanisms based on global information perform better than the ones based on local information, they are only suitable for those networks where frequent network state changes are not expected.

9.2 Routing in WDM Networks

In a wavelength routing scenario, most lightpath control protocols proposed currently in the literature use a source routing mechanism which allows the source node to compute an end-to-end route for the incoming connection.

Unlike a traditional *IP/MPLS* scenario where the routing process only looks for the optimal route, in *WDM* networks the routing process, the *Routing and Wavelength Assignment* problem (*RWA*) [70], must find both the physical nodes and links that configure the lightpath (routing sub-problem), and the wavelength/s to be used on all the links along the lightpath (wavelength assignment sub-problem), in

such a way that network resource utilization is optimised. Therefore, there are two steps involved in the lightpath establishment process. Firstly the network must decide a route for the connection, and secondly reserve a suitable wavelength on each link along the selected route

In general the RWA is addressed differently depending on the availability of wavelength conversion capabilities. Wavelength routed networks without wavelength conversion are known as *wavelength-selective* (*WS*) networks. In such a network, a connection can only be established if the same wavelength is available on all the links between the source and destination nodes (*wavelength-continuity constraint*). This may cause high blocking probability. Wavelength routed networks with wavelength conversion are known as *wavelength-interchangeable* (*WI*) networks. In such networks, each router is equipped with wavelength converters so that a lightpath can be set up using different wavelengths on different links along the route.

There are basically three approaches to dealing with the routing sub-problem: *fixed-routing*, *fixed-alternate routing*, and *adaptive routing*. *Fixed routing* always selects the same pre-computed route for a source-destination pair. In *fixed-alternate routing* a set of fixed pre-computed lightpaths exists for a source-destination pair, and one of them is selected according to a certain heuristic. In *adaptive routing* the lightpath is dynamically selected depending on the current network state, according to a particular heuristic, such as the *shortest path* or the *least-congested path* (*LCP*) [71]. The *LCP* selects those links with the most available wavelengths to carry the lightpath. Notice that approaches based on fixed routes reduce the complexity, but unlike adaptive routing may suffer from higher connection blocking. In general, *fixed routing* is the simplest in implementation while *adaptive routing* produced best global network performance. On the other hand, *fixed-alternate routing* offers a trade-off between computing overhead and network performance.

Once the source node selects a route for the incoming connection, a distributed reservation protocol must be used to reserve the proper wavelength on each link along the selected path. A large number of different heuristics has been proposed for the wavelength assignment sub-problem: Random, First-Fit, Least-Used, Most-Used,

Min-Product, Least-Loaded, Max-Sum and Relative Capacity Loss, for example. These can each be combined with different routing mechanisms.

There are two types of wavelength reservation protocols, the *forward reservation protocol* (*FRP*) and the *backward reservation protocol* (*BRP*) [72]. Figure 56 and Figure 57 illustrate the *FRP* and *BRP* performance respectively. When using *FRP* the source node (*S_N*) sends a request packet (*REQ*) to the destination node (*D_N*) along the selected route. The *REQ* packet tries to reserve a suitable wavelength at the intermediate nodes (*I_Ns*) when available. If succeed the *REQ* packet is forwarded and the wavelength is reserved. Otherwise, a negative acknowledgment (*NAK*) packet is sent back to the source node also dropping the *REQ* packet and releasing already reserved wavelengths along the forwarded *I_Ns*. When *REQ* packet reaches the *D_N*, it sends an acknowledgment packet (*ACK*) back to the source node, configuring the *I_Ns*.

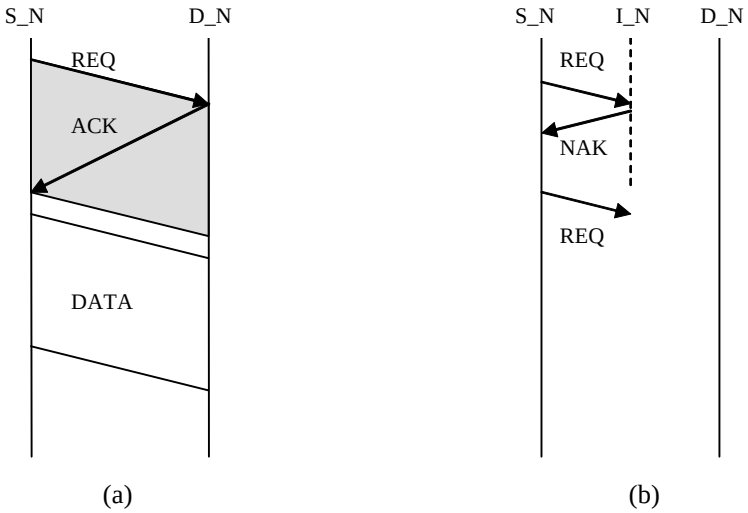


Figure 56. Forward reservation: (a) Successful; (b) Unsuccessful

The lightpath is successfully established when the *ACK* packet reaches the source node. Using this type of reservation protocol introduces a decrease in the wavelength utilization, since wavelengths are reserved on the source node. Using *BRP* improves the wavelength utilization. In fact, shaded areas in Figure 56 and Figure 57 represent the time in which wavelengths are reserved but not used. When using *BRP* a *Probe* (*PROB*) packet is sent by the source node along the selected route to the destination node. This *PROB* packet just collects information about available wavelengths on

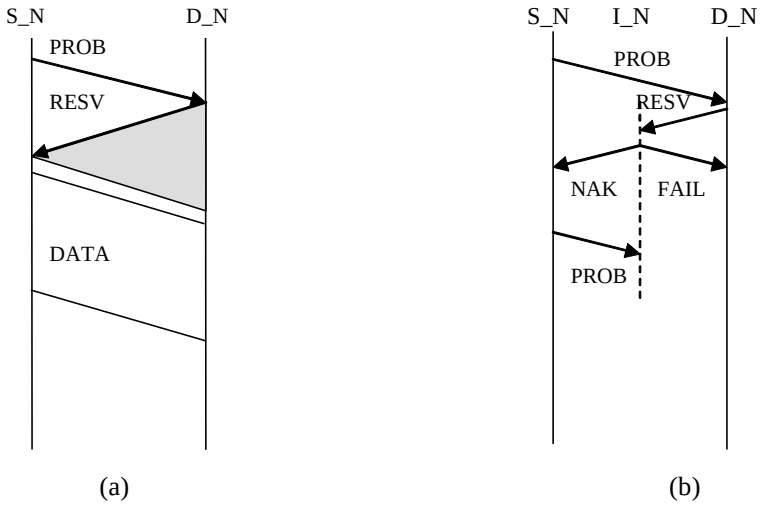


Figure 57. Backward reservation: (a) Successful; (b) Unsuccessful

each node instead of reserving wavelengths. When the *PROB* packet reaches the destination node, it selects a suitable wavelength and sends a reservation (*RESV*) packet back to the source node along the reverse path, which really performs the wavelength reservation on each intermediate node. When the reservation process does not succeed, a failure (*FAIL*) packet is sent to the destination node and a negative acknowledgment (*NAK*) is sent to the source node. While the *NAK* packet only informs the source node about the reservation failure, the *FAIL* packet must release the wavelengths already reserved by the *RESV* packet along the selected path. The lightpath is finally established when the *RESV* packet reaches the source node.

Chapter 10

The Routing Inaccuracy Problem in WDM Networks

As it has been said for an *IP/MPLS* scenario, in large dynamic networks the number of update messages generated by any update mechanism needed to keep network state information correctly updated, may overflow the network with signalling messages, causing an undesirable overhead. In this Chapter we focus on distributed lightpath control under global information, which is more appropriate and reliable for highly dynamic large networks if the network state information perfectly represents the current network state.

10.1 Problem Definition

As mentioned earlier, adaptive distributed routing mechanisms based on global network state information in a dynamic environment require a huge number of update messages to correctly update the network state databases on each node, which implies an undesirable signalling overhead. In order to overcome this signalling

overhead issue, the number of update messages is limited by instituting a triggering policy. An unfortunate effect of limiting the number of update messages is that the information contained in the network state databases does not represent a current picture of the network. Indeed, the *RWA* problem under inaccurate routing information produces an increment in the connection blocking probability [73].

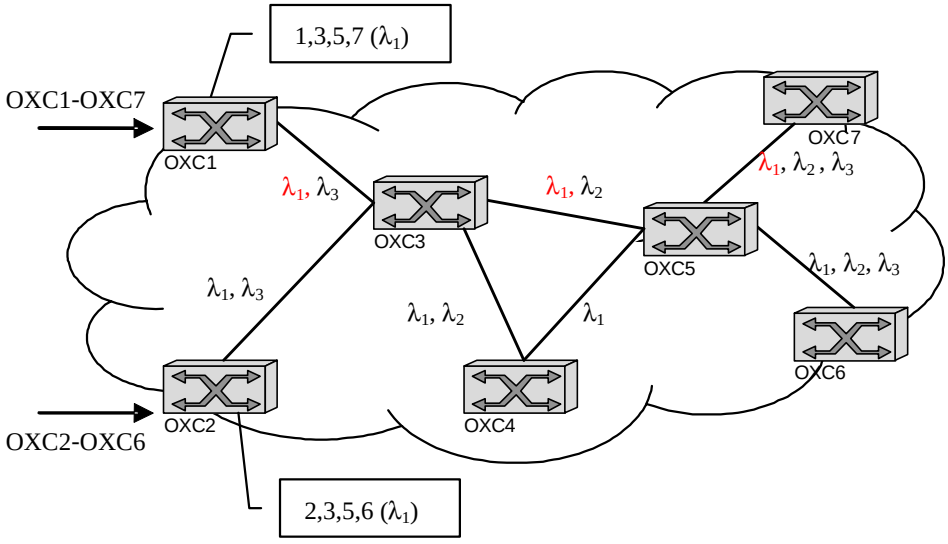


Figure 58. Routing Inaccuracy effects in WDM networks

As previously done in an *IP/MPLS* scenario, Figure 58 illustrates the routing inaccuracy problem. Metric on each link stands for the residual capacity in terms of available wavelengths. Assuming explicit routing, edge nodes computes a physical route and assigns a wavelength reacting to an connection request, so no grooming is considered. When a connection request reaches OXC1 demanding a lightpath from OXC1 to OXC7, OXC1 must select a route and assign a wavelength. This selection is performed based on the network state information contained in its database. Suppose that OXC1 selects the shortest route made up of OXC1, OXC3, OXC5, OXC7 with l_1. If update messages are not triggered, the information contained in the databases on the edge nodes is out-to-date. In red there are those wavelengths read as available on the databases although they are not really available. Suppose that a connection request demanding a lightpath from OXC2 to OXC6 reaches OXC2. OXC2 selects a lightpath based on its network state information, which now is outdated. It is perfectly reasonable a selected lightpath made of OXC2, OXC3,

OXC5, OXC6 with l_1. A set-up message will be sent along the selected route to the destination to establish the lightpath. When receiving the set-up message OXC3 checks the real wavelength availability by looking at its network state database. As the selected l_1 is not available on the output link the set-up message is dropped and so the connection is rejected.

10.2 State of the Art

Regarding *WDM* networks, in [74] the effects produced in the blocking probability because of inaccurate routing information when selecting lightpaths are shown by simulation. The authors indeed verify over a fixed topology that the blocking ratio increases when routing is done under inaccurate routing information. The routing uncertainty is introduced by adding an update interval of 10 seconds. Some other simulations are performed to show the effects on the blocking ratio due to changing the number of fibres on all the links. Finally, the authors argue that new *RWA* algorithms that can tolerate imprecise global network state information must be developed for dynamic connection management in *WDM* networks.

In [75] the routing inaccuracy problem is addressed by modifying the lightpath control mechanism, and a new distributed lightpath control based on destination routing is suggested. The mechanism is based on both selecting the physical route and wavelength on the destination node, and adding rerouting capabilities to the intermediate nodes to avoid blocking a connection when the selected wavelength is no longer available at set-up time in any intermediate node along the lightpath. There are two main weaknesses of this mechanism. Firstly, since the rerouting is performed in real time in the set-up process, wavelength usage deterioration is directly proportional to the number of intermediate nodes that must reroute the traffic. Secondly, the signalling overhead is not reduced, since the *RWA* decision is based on the global network state information maintained on the destination node, which must be perfectly updated.

Another contribution on this topic can be found in [76] where authors propose a mechanism whose goal is to control the amount of signalling messages flooded throughout the network. Assuming that update messages are sent according to a hold-

down timer regardless of frequency of network state changes, authors propose a dynamic distributed bucket-based Shared Path Protection scheme. (an extension of the *Shared Path Protection*, *SPP* scheme presented in [77]). Therefore, the amount of signalling overhead is limited by both fixing a constant hold-down timer which effectively limits the number of update messages flooded throughout the network and using buckets which effectively limits the amount of information stored on the source node, i.e. the amount of information to be flooded by nodes. The effects of the introduced inaccuracy are handled by computing alternative disjoint lightpaths which will act as a protection lightpaths when resources in the working path are not enough to cope with those required by the incoming connection. Authors show by simulation that inaccurate database information strongly impacts on the connection blocking. This increase in the connection blocking may be limited by properly introducing the suitable frequency of update messages. According to the authors, simulation results obtained when applying the proposed scheme along with a modified version of the *OSPF* protocol, may help network operators to determine that frequency of update messages which better maintains a trade-off between the connection blocking and the signalling overhead.

The next Chapter presents a new adaptive source routing mechanism, *BYPASS Based Optical Routing* (*BBOR*) that aims to reduce the connection blocking probability due to performing routing and wavelength assignment decisions under inaccurate routing information.

Chapter 11

BBOR: Adaptation of the BBR Mechanism to WDM Networks

BYPASS Based Optical Routing (*BBOR*) [67] is a new adaptive source routing mechanism, which dynamically computes explicit lightpaths in an *ASON* without wavelength conversion capabilities based on global network state information, aiming to reduce the connection blocking probability due to routing and wavelength assignment decisions performed under inaccurate routing information. The proposal presented in this section modifies the *BBR* structure to make it capable of addressing the effects of having the inaccurate routing information that results from applying a certain triggering policy to reduce the signalling overhead in an ASON.

Although the main concept of *BBOR* is similar to *deflection routing* [78] (studied for packet switched networks) or *alternate-link routing* [79], important differences exist between them. In *alternate-link routing* (an adaptive routing with local information approach), alternate paths are pre-computed and sorted in the routing table of each node based on local network state information and can be used

depending on the resource availability at any time. Instead, based on global network state information, the *BBOR* mechanism only computes *bypass-paths* for those links that potentially might not cope with the traffic requirements, and the usage of these *bypass-paths* is decided at path set-up time depending on the resource availability.

Moreover, in spite of the fact that the *BBOR* mechanism also introduces a rerouting mechanism, unlike the mechanism suggested in [75] the alternative paths are pre-computed at the source node along with the selected lightpath. In this way connection set-up time and wavelength usage deterioration are both reduced.

11.1 BBOR Description

As mentioned above, the source of routing inaccuracy analysed in this Thesis is mainly due to the introduction of a triggering policy in order to reduce the signalling overhead produced by the update messages. Thus, the *BBOR* mechanism includes two main aspects: a triggering policy adapted to the RWA problem to reduce routing signalling, and a routing algorithm based on the dynamic bypass concept to counteract the effects of the routing inaccuracy produced by this routing signalling reduction. It is important to note that the routing algorithm includes both path selection and wavelength assignment. The triggering policy and the feasible routing algorithms inferred from the *BBOR* mechanism are now in detail described.

11.1.1 BBOR: A New Triggering Policy

Existing triggering policies are based on updating by either a periodical refresh or sending an update message whenever there is a change in the network state. In the first case, by modifying the refresh time value, the network state accuracy and the number of update messages can be adjusted. However, this scheme is not valid for large dynamic networks. In the second case, an important signalling overhead is added. In order to improve the update process, the *BBOR* mechanism introduces a new triggering policy based on a threshold value that aims to include network congestion (available network resources) in the triggering decision. In this way, a network node triggers an update message whenever a fixed number *N* of wavelengths changes their status (i.e., after a fixed number of *N* connections are established or

released). By changing the value of *N*, we can evaluate the impact of different degrees of inaccuracy on the connection blocking ratio.

11.1.2 BBOR: A New Routing Algorithm

The main characteristic of the possible routing algorithms included in the *BBOR* mechanism is that they allow several nodes along the selected path to dynamically reroute the set-up message to a different route when, due to the wavelength unavailability produced by computing the selected paths according to inaccurate routing information, this set-up message would be rejected by any one of the intermediate nodes. Two possible rerouting options exist: change the route while maintaining the wavelength, or change the wavelength while maintaining the route. In a *wavelength continuity constraint* scenario, the first one is chosen. Therefore, when an intermediate node decides to reroute the set-up message it sends this message along a different route (*bypass-path*), which bypasses the link that cannot fulfil the *wavelength continuity constraint*.

Any routing algorithm derived from the *BBOR* mechanism consists of three basic processes: (1) decide which wavelength of which link (bundle of *B* fibres) might be bypassed, (2) include these wavelengths as a parameter to be considered when selecting the lightpath, and (3) compute the *bypass-paths*.

Concerning the first process, the wavelengths that have to be bypassed are referred to as *Obstruct-Sensitive-Wavelengths (OSWs)*. The classification of a wavelength l_i as an *OSW* ($l^{os}{}_i$) on a certain link depends on the triggering policy used to update the network state information. *C* being the total number of a certain l_i on a link and *R* being the current number of available (not assigned to an already established lightpath) l_i on this link, then according to the *BBOR* triggering policy described above, this l_i is defined as $l^{os}{}_i$ in this link when *R* is lower than a percentage T_p *(threshold percentage)* of *N*. Note that *N is* the number of wavelength status changes that trigger an update message. Hence, changing the T_p value can modify the granularity in the OSW definition.

Concerning the second process, the source node has to take into account the number of l_i defined as *OSW* in order to properly resolve the *RWA* problem. A new

parameter OSW_i *(L, F)*, where *L* is the number of links where l_i has been defined as *OSW* and *F* is the minimum value of available wavelengths along the lightpath, has therefore been defined. Applying this parameter, two different algorithms can be inferred from the *BBOR* mechanism, *ALG1* and *ALG2*. *ALG1* lies in selecting those $l_i s$ in all the links of the shortest paths (minimum number of hops) that minimize *L* in $OSW_i(L,F)$. If more than one wavelength is compliant with this condition, the algorithm selects the less congested by checking the *F* value in $OSW_i(L,F)$. *ALG2* lies in selecting the less congested $l_i s$ on the shortest paths according to the *F* value in $OSW_i(L,F)$. If more than one wavelength is compliant with this condition, the algorithm selects that l_i which minimizes the *L* value in $OSW_i(L,F)$. In other words, *ALG1* prioritises minimizing the number of obstructions or bottlenecks while *ALG2* prioritises minimizing the congestion.

Concerning the third process, once the lightpath has been selected a *bypass-path* must be computed for those wavelengths defined as *OSW* in this lightpath, in such a way that the *wavelength continuity constraint* is guaranteed. Although other criteria could be used to compute the *bypass-paths* (this is left for further studies), such as minimizing the number of wavelengths defined as *OSW*, the shortest (minimum number of hops) *bypass-paths* are selected. In order to simplify the *bypass-path* computation, when a *bypass-path* exists on a link for a particular l^{os}_i, this path will also be used as the first option to bypass any other l^{os}_j on this link. Summarizing, in order to explicitly distribute the *bypass-paths* in the set-up message, source nodes must both detect those wavelengths on a link that potentially will not be available when establishing the path, and compute a *bypass-path* for each one of these wavelengths. A brief description of the *BBOR* mechanism is presented in Figure 59.

11.2 Example Illustrating How BBOR Works

The topology shown in Figure 60 is used to illustrate how *BBOR* works. Considering that every OXC includes control functions with signalling capabilities, we assume *C* = 10 fibres per link and 4 wavelengths per fibre. Update messages are

BYPASS BASED OPTICAL ROUTING MECHANISM

Input: An incoming connection request between a source-destination pair *(s,d)* with a *wavelength continuity constraint*

***Output*:** An explicit route from *s* to *d* with a common available wavelength on all the links along the path and with enough *bypass-paths* to bypass the routing inaccuracy effects in the *obstruct-sensitive wavelengths*.

Algorithm:

1. Select the shortest paths
2. Mark those wavelengths that are defined as *OSW*
3. Depending on the algorithm to be used, ALG1, ALG2:

   ALG1:
   - Select that l_i on all the paths minimizing the *L* value in $OSW_i(L,F)$
   - If more than one exists the less congested is selected according to the *F* value in $OSW_i(L,F)$.

   ALG2:
   - Select the less congested wavelength on each path according to the *F* value in $OSW_i(L,F)$.
   - If more than one exists, select that l_i on all the paths minimizing the *L* value in $OSW_i(L,F)$.
4. Compute a *bypass-path* for all wavelengths defined as *OSW*.
5. Decide which *bypass-paths* must be used in accordance with real available resources in the path setup time

Figure 59. *BBOR* description

sent according to $N = 6$ and a wavelength l_i is defined as OSW_i according to T_p = 50% (i.e., when the minimum number of available wavelengths on this link is lower than or equal to 3). Incoming connection requests arrive between OXC1-OXC4.

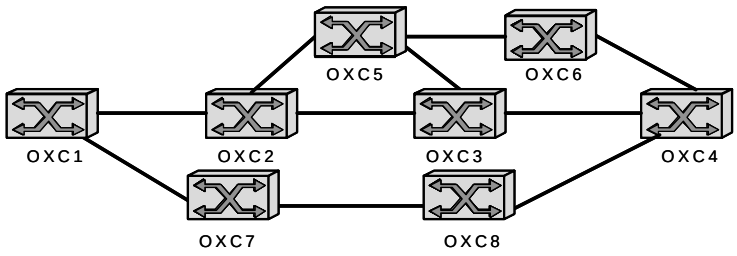


Figure 60. Network topology used in the *BBOR* illustrative example

In Table 8 the network state information existing in OXC1 is shown. It represents the number of available wavelengths for all the links. According to this information, Table 9 shows the routing table existing in OXC1, where all the feasible lightpaths between OXC1 and OXC4 are pointed out. In addition, the minimum number of available wavelengths and the OSW_i *(L,F)* parameter are shown for each lightpath.

Table 8. Network State in OXC1

Link (OXC)	l_1 l_2 l_3 l_4	*Link (OXC)*	l_1 l_2 l_3 l_4
1-2	6 3 3 6	5-6	0 7 3 3
2-3	2 3 6 0	6-4	1 1 1 1
3-4	6 3 0 2	1-7	6 3 1 6
2-5	6 2 0 1	7-8	0 3 6 1
5-3	6 6 6 6	8-4	6 6 0 6

Table 9. Routing Table in OXC1

Route (OXC)	l_1 l_2 l_3 l_4	*OSW$_i$ (L,F)*
1-2-3-4	2 3 0 0	$\lambda_1(1,2)$, $\lambda_2(3,3)$
1-2-5-3-4	6 2 0 1	$\lambda_2(3,2)$, $\lambda_4(2,1)$
1-2-5-6-4	0 1 0 1	$\lambda_2(3,1)$, $\lambda_4(3,1)$
1-7-8-4	0 3 0 1	$\lambda_2(2,3)$, $\lambda_4(1,1)$

Finally, Table 10 shows, hop-by-hop, the process of applying the *BBOR* mechanism. As a result, a different lightpath and a different wavelength are selected to transmit the traffic depending on the algorithm in use. Thus, l_1 along the path made of OXCs 1-2-3-4 and l_2 along the path made of OXCs 1-7-8-4 are selected by *ALG1* and *ALG2* respectively. In addition, since l_1 is defined as OSW_1 on link OXC2-OXC3 a *bypass-path* through OXCs 2-5-3 is also selected.

Table 10. Illustrative Example

BBOR steps	*Algorithm 1 (ALG1)*	*Algorithm 2 (ALG2)*
1	path 1: 1-2-3-4 path 2: 1-7-8-4	path 1: 1-2-3-4 path 2: 1-7-8-4
2 (ALG1)	path 1: λ_1(**1**,2) path 2: λ_4(**1**,1)	path 1: λ_2(3,**3**) path 2: λ_2(2,**3**)
3 (ALG1)	**path 1:** λ_1(1,**2**) path 2: λ_4(1,**1**)	path 1:λ_2(**3**,3) **path 2**:λ_2(**2**,3)
4	λ_1 is *OSW* on 2-3 *bypass-path*:2-5-3	λ_2 is *OSW* on 1-7-8 No *bypass-path*

However, when using *ALG2*, path 2 and l_2 are the *RWA* result. In this case, l_2 is OSW_2 on links OXC1-OXC7-OXC8. It is not possible to find a proper *bypass-path* to

directly bypass these links. In this case, *BBOR* cannot be completely applied. Further extensions must be added to the *BBOR* mechanism to cope with this problem.

11.3 Performance Evaluation

In this section the simulation scenario in which the *BBOR* mechanism has been evaluated, the parameters used to test its benefits and the obtained results are presented. However, before evaluating the proposed mechanism, the effects of applying the *BBOR* mechanism over the time needed to set-up a lightpath are analysed later on.

The time needed to set-up a lightpath is defined as the time taken from the moment an incoming connection request reaches the source node to the moment the lightpath is successfully established. This time depends on:

T_c = Time taken by the source node to compute a route
T_{c_b} = Time taken by the source node to compute a *bypass-path* route
n_s = Number of hops in the shortest path
n_{OS} = Number of wavelengths defined as *Obstruct-Sensitive* in the selected route
m = Number of wavelengths that really are not available in any intermediate node along the selected route
n_{bi} = Number of hops in the *bypass-path* i
T_d = Propagation delay on each link
T_p = Time taken by an intermediate node to process a connection request
T_r = Time taken by a node to reserve a wavelength

The set-up message sent by the source node takes a time t_d to reach the destination node. This time depends on the number of wavelengths defined as *OSW*. Thus, we define T_S as the total time needed to establish the connection, the two-way delay needed to establish a lightpath. Different cases can be analysed depending on the number of *OSW*:

1) There are no wavelengths defined as *OSW*.

$$T_s = T_c + 2 \text{ x } n_s \text{ x } T_d + (2 \text{ x } n_s + 1) \text{ x } T_p + (n_s + 1) \text{ x } T_r \qquad (7)$$

2) There are n_{OS} wavelengths defined as *OSW* but none are used

$$T_s = T_c + T_{c\_b} \text{ x } n_{OS} + 2 \text{ x } n_s \text{ x } T_d + (2 \text{ x } n_s + 1) \text{ x } T_p + (n_s + 1) \text{ x } T_r \qquad (8)$$

3) There are n_{OS} wavelengths defined as *OSW* and *m* are used, where

$$m \subset n_{OS} \quad \text{and} \quad m \leq n_{OS} \tag{9}$$

Now the time T_s can be represented as:

$$\begin{aligned} T_s &= T_c + T_{c\_b} \text{ x } n_{OS} + 2 \text{ x } \left[(n_s - m) + \sum_{i=1}^{m} n_{bi} \right] \text{x } T_d + \\ &\left[2 \text{ x } \left[(n_s - m) + \sum_{i=1}^{m} n_{bi} \right] + 1 \right] \text{x } T_p + \quad + \left(n_s - m + 1 + \sum_{i=1}^{m} n_{bi} \right) \text{x } T_r \end{aligned} \tag{10}$$

Although the *BBOR* mechanism requires an increment in the time needed to set up a lightpath compared to another mechanism that does not compute *bypass-paths*, this time does not substantially affect wavelength usage. This claim is next clarified by applying the above-described equations to the network topology of Figure 60. Using *ALG1*, λ_1 on OXC1-OXC2-OXC3-OXC4 represents the selected lightpath. This wavelength is defined as *OSW* in the link OXC2-OXC3. Three different cases are analysed. Firstly, we compute the time needed to establish the lightpath when no *BBOR* mechanism is applied. Therefore, $n_S = 3$, and the T_S is

$$\begin{aligned} T_s = T_c + 2 \text{ x } n_s \text{ x } T_d + (2 \text{ x } n_s + 1) \text{ x } T_p + (n_s + 1) \text{ x } T_r = \\ T_c + 6T_d + 7T_p + 4T_r \end{aligned} \tag{11}$$

Secondly, we compute the time needed to establish the lightpath when applying the *BBOR* mechanism but the *bypass-path* computed to bypass the link OXC2-OXC3 is not really used when the set-up message reaches OXC2. Therefore, $n_{OS} = 1$, $n_S = 3$ and T_S is

$$\begin{aligned} T_s = T_c + T_{c\_b} \text{ x } 1 + 2 \text{ x } 3 \text{ x } T_d + (2 \text{ x } 3 + 1) \text{ x } T_p + (3 + 1) \text{ x } T_r = \\ T_c + T_{c\_b} + 6T_d + 7T_p + 4T_r \end{aligned} \tag{12}$$

Lastly, we represent the time needed to establish the lightpath when the *bypass-path* computed to bypass the link OXC2-OXC3 is used. The final end-to-end lightpath is made of OXC1-OXC2-OXC5-OXC3-OXC4. Therefore, $n_{OS} = 1$, $n_S = 3$, $m = 1$, $n_{bi} = 2$ and T_S is

$$T_s = T_c + T_{c\_b} \text{ x } 1 + 2 \text{ x } \left[(3-1) + \sum_{i=1}^{1} 2 \right] \text{x } T_d + \left[2 \text{ x } \left[(3-1) + \sum_{i=1}^{1} 2 \right] + 1 \right] \text{x } T_p + \left(3-1+1+\sum_{i=1}^{1} 1 \right) \text{x } T_r = T_c + T_{c\_b} + 8T_d + 9T_p + 5T_r \quad (13)$$

It can be seen that the increment of time introduced due to applying the *BBOR* mechanism when no *bypass-paths* are used is just the time needed to compute these *bypass-paths*. Moreover, as the time increment does not affect the time in which a certain wavelength is reserved but not used (since it is computed before sending the set-up message), this does not produce network inefficiency. As far as comparing the first and the last situation, the increment generated in the path set-up time can be represented as

$$\Delta T_s = T_{c\_b} + 2T_d + 2T_p + T_r \quad (14)$$

It can be observed that only the time needed to propagate, process and reserve a wavelength affects the time in which a wavelength is reserved but not used. However, this increment, proportional to the number of *bypass-paths* to be computed is very low.

Once the impact of the *BBOR* mechanism on the lightpath set-up time has been analyzed, results obtained when evaluating the *BBOR* mechanism by simulation are discussed. The simulations are performed over the network topology shown in Figure 61 where the possible source-destination pairs are randomly selected. We suppose a 5-fibre topology, with 16 wavelengths on each fibre on all the bi-directional links.

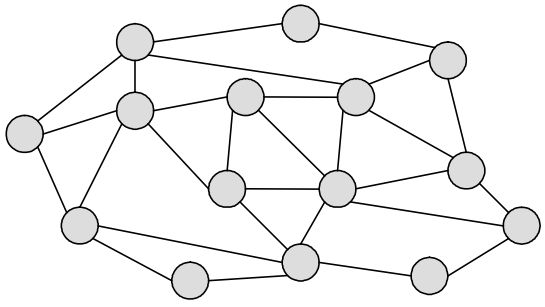


Figure 61. Topology used in simulations

Connection arrivals are modelled as a Poisson distribution with arrival rate ? and the connection holding time is assumed to be exponentially distributed with average value ($1/m$). Assuming adaptive routing and without loss of generality, routes are computed after applying the shortest path algorithm.

Three routing algorithms are evaluated by simulation: *First-Fit*, *ALG1* and *ALG2*. *First-Fit* is that wavelength assignment heuristic which selects the lowest numbered available wavelength among a set of numbered wavelengths. In the next figures the effects produced in the network performance by applying the *BBOR* mechanism are shown: the reduction in the number of update messages when the triggering policy defined in the *BBOR* mechanism is applied, and the blocking probability reduction obtained when applying the *BBOR* mechanism. Both effects are analysed as a function of both *N* (number of wavelength state changes that trigger an update message) and T_p (threshold percentage of *N* which defines when a wavelength is defined as *OSW*) values.

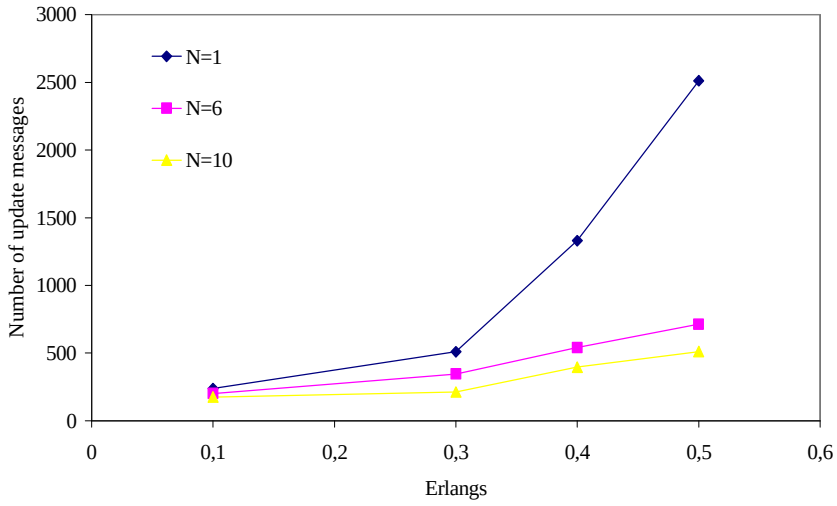


Figure 62. Number of update messages

Figure 62 shows the reduction obtained in the quantity of update messages supplied to the network when increasing the values of *N*. As expected, the larger the *N* the lower the number of update messages. Note that the case of $N = 1$ corresponds to a policy that triggers update messages whenever a change occurs.

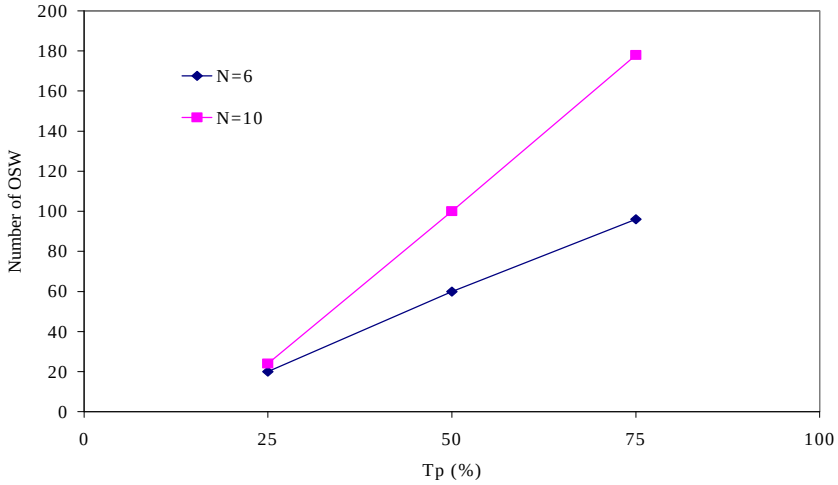


Figure 63. Number of *OSW* as a function of the threshold percentage T_p value

Figure 63 shows the number of wavelengths defined as *OSW* as a function of the T_p value. The number of defined *OSW*s grows with the T_p value, since the minimum number of available wavelengths on a certain link used to define when a wavelength is an *OSW* on this link is also directly proportional to the T_p value.

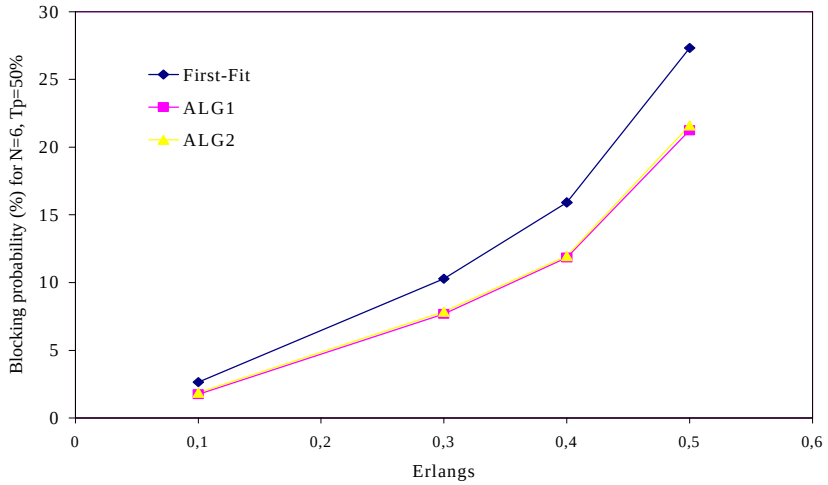


Figure 64. Blocking probability for $N = 6$ and $T_p = 50\%$

According to the results obtained in Figure 63 the blocking probability is evaluated considering a value of $T_p = 50\%$. Figure 64 compares the blocking probability obtained by the *BBOR* algorithms, and the shortest path algorithm combined with the *First-Fit* approach, considering a value of $N = 6$. It can be seen

that in the worst case a blocking probability reduction of 6.08% is obtained when applying the *BBOR* mechanism.

It is worth to notice that the lightpath holding time does not affect on the connection blocking since update messages keep up with the frequency of network changes so regardless of any hold-down timer. Really, the lightpath holding time only leads to increment the number of update messages flooded throughout the network.

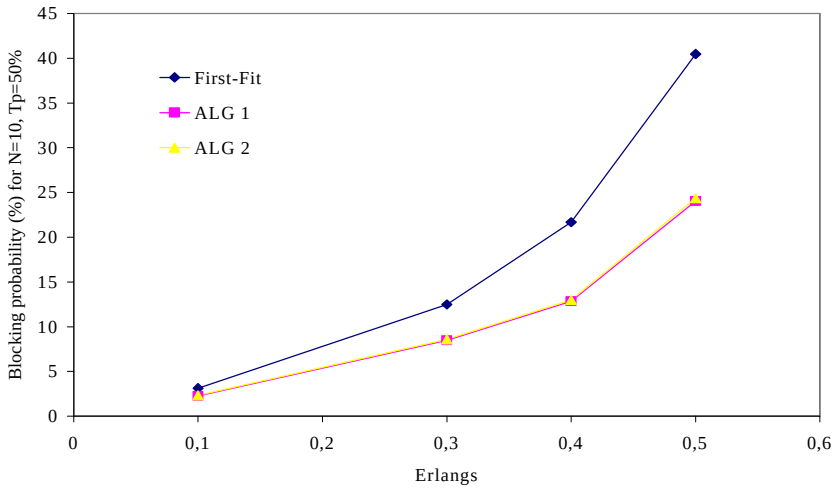


Figure 65. Blocking probability for N = 10 and T_p = 50%

Analogously, the blocking probability for N = 10 is shown in Figure 65. In this case, the blocking probability reduction achieved by the *BBOR* algorithms compared to the *First-Fit* heuristic reached 16.12%.

Analysing a fixed blocking probability value (27.32%) for the *First-Fit* heuristic shows that unlike Figure 64, where a reduction of 6.08% with N = 6 is obtained, in Figure 65 where N = 10, the reduction is about 11%. Therefore, according to the obtained simulation results, the *BBOR* mechanism obtains the largest blocking probability reduction when the N value increases, that is, when the number of update messages has been reduced as well.

As a conclusion it is possible to say that the *BBOR* mechanism reduces both signalling overhead and the negative effects produced by having inaccurate routing information.

Chapter 12

The BBOR Mechanism in a Wavelength Conversion Scenario

There are plenty of references in the literature where the need of adding conversion capabilities to the network is deeply justified. This Chapter extends the *BBOR* mechanism to be applied to networks with conversion capabilities, that is, *WI* networks [68].

12.1 Wavelength Interchangeable Networks

In *Wavelength Interchangeable Networks* (*WI*), also called *Wavelength Convertible Networks*, lightpaths may be selected without using the same wavelength in all the links along the selected lightpath. As a consequence, the global network efficiency is largely improved. If a wavelength converter provides the ability to translate any input wavelength to any output wavelength, i.e., full range conversion, and every node of the network includes a wavelength converter, the network is defined as having full wavelength-conversion capabilities. In this case, the network is

equivalent to a circuit switched network, where only the routing subproblem must be considered. However, the cost associated to provide a wavelength converter at every node is currently not affordable. Therefore, other solutions based on limiting the global wavelength conversion in a network appear to design a *WI* network. There are three main issues to be considered. First, the global conversion capability may be reduced by having only a few nodes with conversion capabilities, i.e. sparse conversion, modeled by the conversion density *q* of the network. Second, converters may be shared among various output ports of a node. Third, the range of wavelength conversion is limited to a fixed value *k*, i.e., limited range wavelength conversion, defining the *degree of translation D* as

$$D = \frac{100\,k}{\Lambda - 1}\,(\%) \qquad (15)$$

where Λ is the total number of wavelengths on a link.

In this way an input wavelength l_i may only be translated to wavelengths $l_{max(i-k,1)}$ through $l_{min(i+k,?)}$. It is show in [80] that a substantial improvement in the global blocking probability of the network when limited-range wavelength converters with as little as 25% of the full conversion range are introduced.

In this Chapter the impact on the blocking probability because of applying the *BBOR* mechanism to a network with wavelength conversion capabilities are in detail analyzed. A new routing algorithm is generated. The path selection process used in the new suggested routing algorithm is modified regarding the *ALG1* and *ALG2* already proposed in Chapter 11.

12.2 ALG3: Applying the BBOR Mechanism to WI Networks

A new algorithm inferred from the *BBOR* mechanism, named *ALG3* (as an extension of the already proposed *ALG1* and *ALG2*), is now suggested to address the routing inaccuracy problem in a wavelength conversion scenario. *ALG3* incorporates several different aspects in comparison to the *ALG1* and *ALG2*. In fact, although the main concepts are the computation of both the OSW_i *(L,F)* parameter and the *bypass-paths*, in *ALG3* both aspects are differently handled. There are three main differences between *ALG3* and the other ones:

Firstly, *ALG3* does not select only the shortest paths. Instead, the K-shortest paths of all possible disjoint paths between source and destination nodes are computed. Secondly, unlike *ALG1* and *ALG2* where the weight of each link was separately defined by the attributes *L* and *F* of the $OSW_i(L,F)$ parameter, in *ALG3*, the weight associated to each link is represented by the factor *L/F*. This factor stands for a balance between the number of potentially obstructed links and the real congestion instead of choosing one against the other. Moreover, since longer paths than the shortest ones can be selected, the length of the path is also included in the path decision. Hence, in order to avoid those paths that are either widest (in terms of wavelength availability) but too long or shortest but too narrow, the weight factor of each path is modelled by F_p according to the expression

$$F_p = n\left(\frac{L}{F}\right) \tag{16}$$

where *n* is the number of hops in the selected path.

Finally, once the path has been selected, *bypass-paths* are computed. Now, before computing the *bypass-paths* it is necessary to know whether the output link where a certain l_i is defined as l^{os}_i belongs to a node with conversion capabilities. If it does, the bypass dynamic concept can be simply modelled by converting the wavelength. If it does not or there are not available wavelengths where limited conversion can be done, the *bypass-paths* are computed similarly to *ALG1* and *ALG2*. The box enclosed in

Figure **66** shortly summarizes *ALG3*.

12.3 Performance Evaluation

To evaluate the *BBOR* mechanism in *WI* networks a set of simulations have been carried out over the network topology shown in Figure 61, where the possible source-destination pairs are randomly selected. We suppose a 5-fibre topology, with 10 wavelengths on all the fibres on all the bi-directional links. Connection arrivals are modelled by a Poisson distribution and the connection holding time is assumed to be exponentially distributed. Each arrival connection requires a full wavelength on each link it traverses.

BYPASS BASED OPTICAL ROUTING (ALG3)

Input*: An incoming connection request between a source-destination pair (s,d).*

Output*: An explicit route from s to d without fulfilling the wavelength continuity constraint and with enough bypass-paths to bypass the routing inaccuracy effects in the obstruct-sensitive wavelengths.*

Algorithm:

1. Select the k-shortest paths
2. Mark those wavelengths that are defined as OSW
3. Select that l_1 that minimizes the cost according to

$$F_p = n\left(\frac{L}{F}\right)$$

4. Compute the bypass-paths for all wavelengths defined as OSW, considering wavelength conversion when it is possible
5. Decide which bypass-paths must be used in accordance with real available resources in the path set-up time

Figure 66. *ALG3* description

In order to check the benefits obtained when applying *ALG3*, in Figure 67 we firstly compare *ALG1*, *ALG2*, *ALG3* and *First-Fit* algorithms behaviour in an optical network without conversion capabilities by measuring the impact on the blocking probability.

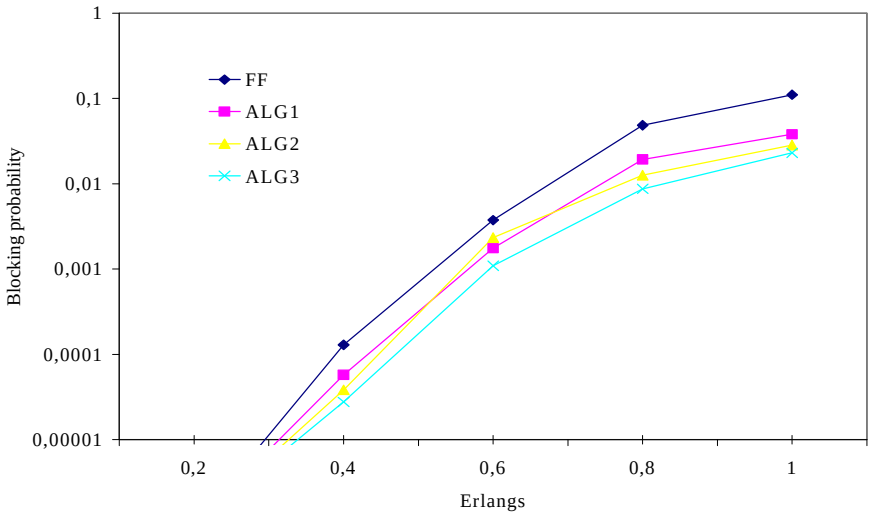


Figure 67. Blocking probability in *WS* networks

According to [4] all the simulations have been performed considering N = 6 (threshold value for triggering updating messages) and T_p = 50% (threshold

percentage of *N* used to define *OSWs*). A light improvement in the blocking probability is obtained with *ALG3* in comparison with *ALG1* and *ALG2*. Actually, although in this scenario no conversion is allowed in the *bypass-path* computation, the weight factor modification implemented in *ALG3* leads to an even more blocking reduction.

Then, Figure 68 exhibits the *ALG3* performance when it is applied to a network with sparse and limited wavelength conversion. In our simulations we consider a fixed value of D = 25% and q in the range of 10%, 25% and 50%. A main aspect to be solved is which nodes should have conversion capabilities. We address this aspect by locating the wavelength converters in those nodes that support more traffic. These nodes are found after running *ALG3* considering there is not wavelength conversion availability in the network. *ALG3* and the *Shortest Path (SP*) algorithms are compared, combining the D and q values. We can see that going on the same trend, *ALG3* also decreases the blocking probability when incrementing the number of conversion capable nodes in the network. Moreover, we can see that when using *ALG3*, increasing the converters density q more than 25% does not imply a significant blocking probability reduction.

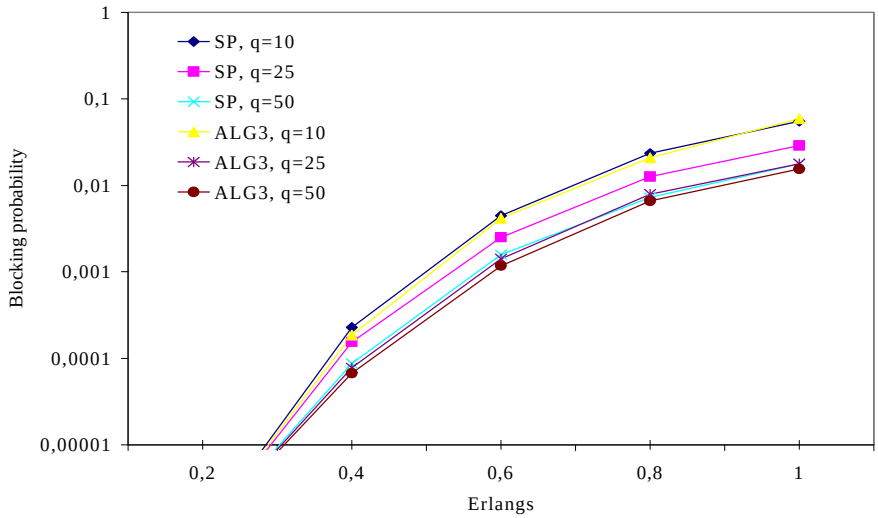


Figure 68. Blocking probability in *WI* networks

After carefully observing Figure 67 and Figure 68 we notice that *ALG3* in a non wavelength conversion scenario presents a similar behaviour than that obtained for

the *SP* algorithm in a wavelength convertible scenario for q = 10% and D = 25%. So, we can say that by applying the *ALG3* a cost reduction can be achieved maintaining the same blocking probability.

Finally comparing the obtained results in *WS* and in *WI* networks, we can argue that *ALG3* can be used as an alternative solution (software solution) to reduce the blocking probability in a *WS* network to that solution based on simply adding wavelength conversion capabilities (hardware solution) to the network. Therefore, taking into account the current high prices for wavelength converters at network elements, *ALG3* is presented as a good solution to reduce the blocking probability while tempering the signalling overhead produced by the update messages

As a summary, in this Part the *BBR* mechanism has been extended to be applied to *WDM* networks. The *BBOR* mechanism addresses the negative effects on global network performance when selecting lightpaths based on inaccurate network state information. Basically, the mechanism *BBOR* lies in two main concepts, a new triggering policy based on a threshold standing for the number of changes and a new routing mechanism. *BBOR* is initially applied to *WS* networks, i.e., networks without wavelength conversion capabilities. Two routing algorithms are inferred from the *BBOR* mechanism, *ALG1* and *ALG2*. These routing algorithms are evaluated by simulation over a certain network topology. Simulation results show the benefits obtained by these algorithms in terms of bandwidth blocking reduction compared to the shortest path combined with the First-Fit heuristic.

Then, the *BBOR* mechanism is applied to *WI* networks, i.e., networks providing conversion capabilities. To simplify the network scenario, sparse and limited conversion is considered. This means that conversion capabilities are not allowed on each node along the network and that nodes having conversion capabilities can convert an input wavelength only to a fixed number of output wavelengths respectively. A new routing algorithm, *ALG3* is evaluated by simulation in comparison with shortest path (*SP*) algorithm. Results obtained by simulation show that *ALG3* in a *WS* network presents similar network performance than that obtained by the *SP* in a *WI* network. Therefore, the *BBOR* mechanism is an excellent option to

reduce the bandwidth blocking without incrementing the unaffordable economical cost because of introducing wavelength converters.

PART IV

CONCLUSIONS AND FUTURE WORK

Main concepts and contributions proposed in the Thesis as well as obtained results are summarized in next Chapters. Before concluding the Thesis, lines of future work where proposed mechanisms may be extended are briefly stated.

Chapter 13

Final Conclusions

The problem known as routing inaccuracy problem arises when the information contained in the network state databases does not perfectly represent a current picture of the network. Many factors, such as non-negligible propagation delay, frequency of the updates, and hierarchical topology aggregation can affect the precision of the network state information, an immediate effect being an increase in the connection blocking probability.

This Thesis deals with mitigating the effects of routing inaccuracy in both *IP/MPLS* and *WDM* networks. The first contribution of this Thesis is the proposal of a new dynamic source QoS routing mechanism, called *BYPASS Based Routing* (*BBR*), which substantially improves global network performance in an *IP/MPLS* scenario. The most significant contribution of this mechanism is the dynamic bypass concept, which lies in applying the idea of path protection, used for fast rerouting after a failure, to the ordinary path selection routing process. Four different algorithms are inferred from the *BBR* mechanism. Two of them, *SOSP* and *OSSP*,

appear when combining the *BBR* mechanism with the shortest path algorithm. The other two, *WSOSP* and *BOSP*, were designed to optimise bandwidth allocation in the routing process. Therefore, the main novelty of the proposed QoS routing mechanism is that it aims at dealing both with the network resource utilization problem (in terms of bandwidth consumption) and the routing information inaccuracy problem.

These algorithms were evaluated over both small and large networks under increasing traffic conditions, and while applying two different triggering policies, namely the Threshold and the Exponential class triggering policies. The large network used for the simulations is a common *ISP* network used in many QoS routing studies in order to provide results with realism. Under these conditions, the *BBR* mechanism was compared with the *SP* algorithm (non-QoS routing), the *WSP* algorithm (QoS routing), and the *SSP* algorithm (an existing solution for the routing inaccuracy problem). As a result it was found that the lowest blocking probability was obtained when applying the *BOSP* algorithm, which under worst conditions achieves a blocking probability reduction of 8.3% compared to the *SSP* algorithm, followed by the *SOSP* algorithm with a reduction of 6 % also compared to the *SSP* algorithm. The *BBR* mechanism is also extended with the *BYPASS Discovery Process* (*BDP*). The *BDP* is introduced to improve the *BBR* applicability by allowing *bypass-paths* to be computed in most network topologies. Obtained results show a blocking probability reduction when including the *BDP* in the *BBR* mechanism.

The main benefit obtained when applying the *BBR* mechanism is the reduction in the bandwidth blocking. However, as dynamic rerouting along pre-computed *bypass-paths* is the solution introduced by the dynamic bypass concept, the obtained reduction depends on the number of *bypass-paths* that can be computed per route. Several simulations are performed to show the impact on the blocking probability as a function of the number of computed *bypass-paths* per route. Results exhibit that larger blocking reduction is obtained when *bypass-paths* are not limited. However the cost introduced is not affordable. A trade-off exists between the bandwidth blocking reduction and the cost. A reasonable cost is reached when three *bypass-paths* per route are allowed. It is also analysed a different policy to reduce the cost. Unlike the above mentioned policy where the cost is reduced by limiting the number of computed *bypass-paths* per route to a fixed value, the new policy limits the cost

by defining the *n_bp* value as a function of the network load, so-called network load dependent. After comparing both policies we can conclude that while a significant reduction in the cost of about 8.6% is achieved when *n_bp* is dependent on the network load, the use of this policy does not involves a significant decrease in the obtained bandwidth blocking when using the other policy.

A second contribution of this work is to apply the *BBR* mechanism suggested for *IP/MPLS* networks to *WDM* networks. The *BYPASS Based Optical Routing* (*BBOR*) mechanism, also based on the dynamic bypass concept, has been proposed. *BBOR* consists of two main components, a triggering policy adapted to optical networks and two routing algorithms, *ALG1* and *ALG2*. The evaluation of *ALG1* and *ALG2* was done by applying them to a wavelength-selective network and comparing their performance with the *First-Fit* heuristic. The simulation results showed a significant reduction in the number of update messages (79.65%) achieved when applying the triggering policy defined in the *BBOR* mechanism. Moreover, analysing the worst case (i.e., under conditions of high volume traffic and routing inaccuracy), both *ALG1* and *ALG2* achieve a similar reduction of about 16% in the connection blocking probability compared to the *First Fit* heuristic.

After that the *BBOR* is applied to networks with conversion capabilities. It is well known that although wavelength converters are currently very expensive, they substantially improve global network performance, such as reducing the connection blocking. This solution could be referred as “hardware solution”. *ALG3* is a new algorithm inferred from the *BBOR* mechanism which computes lightpaths according to a new defined factor which represents a balance between the number of links where the selected wavelength potentially might be unavoidable and the real congestion. *ALG3* achieves a lower connection blocking when is applied to networks with conversion capabilities.

Moreover, *ALG3* has also been applied to networks without conversion capabilities. Obtained results are compared to those obtained when applying the shortest path algorithm to a network with sparse and limited conversion. After carefully analysing these results, the *BBOR* mechanism can be though as a “software

solution" to reduce the connection blocking without requiring an important investment in wavelength converters.

Chapter 14

Future Work

The work and the contributions proposed in this Thesis may be extended given rise to several lines of future work. As stated before, this Thesis proposes QoS routing mechanisms which can be applied to both *IP/MPLS* and *WDM* networks. In these different scenarios there are various ways in which this work may be continued.

An initial extension of the already proposed mechanisms concerns to *IP/MPLS* networks. The network scenario considered so far is based on a flat topology. This network scenario may be modified considered hierarchical routing. In this case, the routing inaccuracy problem is motivated by both the existence of triggering policies and the state aggregation process performed in any hierarchical structure. An analysis of the *BBR* behaviour in this hierarchical network scenario may be performed. Depending on obtained results, new algorithms must be generated.

Up to now, the mechanisms proposed to address the routing inaccuracy problem tackle independently the problem. In fact the *BBR* is applied in *IP/MPLS* networks

and the *BBOR* in *WDM* networks. However both technologies can interoperate in such a way that routing may be simultaneously performed. The implications of this interlayer operation on the routing inaccuracy problem must be carefully analyzed.

All simulation results obtained so far for *WDM* networks focus on reducing the blocking probability ratio. The main goal is to apply a certain wavelength assignment heuristic and to find the reduction obtained in the blocking probability when considering the routing inaccuracy problem. Another future work is related to network dimensioning. In this new scenario, the input parameter is a certain degree of service which is required by the incoming connections. The main objective is to dimension both the number of wavelength converters that must be added to the network and the degree of conversion of each converter, to meet the required degree of service according to traffic conditions.

Finally, the hierarchical network structure can be also applied to *WDM* networks. The impact of different aggregation schemes in the routing inaccuracy and the blocking probability must be analyzed. New routing algorithms taking into account the inaccuracy introduced by the state aggregation process and the different proposed aggregation schemes must be sought.

References

[1] D.O.Awduche, “Requirements for Traffic Engineering over MPLS”, RFC 2702, September 1999.

[2] E.Rosen, A.Viswanathan, R.Callon, “Multiprotocol Label Switching Architecture”, RFC 3031, January 2001.

[3] J. Moy, “OSPF Version 2”, RFC 2328, April 1998.

[4] G. Malkin, “RIP Version 2”, RFC 2453, November 1998.

[5] R. Callon, “Use of IS-IS for Routing in TCP/IP and Dual Environments”, RFC 1195, Dec. 1990.

[6] Y. Rekhter, T. Li, “A Border Gateway Protocol 4 (BGP-4)”, RFC 1711, March 1995.

[7] T. Anjali, C. Scoglio, J. de Oliveira, L.C. Chen, I.F. Akyldiz, J.A. Smith, G. Uhl, A. Sciuto, “A New path Selection Algorithm for MPLS networks Based on Available Bandwidth Estimation”, in Proceedings of QofIS 2002, pp.205-214, Zurich, Switzerland, October 2002.

[8] R. Braden, et al., “Resource Reservation Protocol (RSVP)-Version 1, Functional Specification”, RFC 2205, September 1997.

[9] S. Blake, et al, “An Architecture for Differentiated Services”, RFC2475, December 1998.

[10] K. Nickols, et al., “Definition of the Differentiated services Field (DS Field) in the IPv4 and IPv6 Headers”, RFC2474, December 1998.

[11] D. Awduche, et al. ”RSVP-TE: Extensions to RSVP for LSP Tunnels”, Internet draft<draft-ietf-mpls-rsvp-lsp-tunnel-05.txt>, February 2000.

[12] L. Anderson, et al. “LDP Specification”, RFC 3036, January 2001.

[13] B. Jamoussi, et al., “Constraint-Based LSP Setup using LDP”, RFC3212, January 2002.

[14] J.M. Jaffe, “Algorithms for finding paths with multiple constraints”, Networks 14, pp. 95-116, 1984.

[15] A. Iwata, R. Izmailov, D.-S. Lee, B. Sengupta, G. Ramamurthy and H. Suzuki, “ATM Routing Algorithms with Multiple QoS requirements for Multimedia Internetworking”, IEICE Transactions and Communications E79-B, no. 8, pp. 999-1006, 1996.

[16] P. Van Mieghem, H. De Neve and F.A.Kuipers, “Hop-by-Hop Quality of Service Routing”, Computer Networks, vol.37/3-4, pp. 407-423, October 2001.

[17] S. Chen, K. Nahrstedt, “On Finding Multi-Constrained Paths”, in Proceedings of 7th IEEE International Conference on Communications, Atlanta, GA, pp.874-879, June 1998.

[18] T. Korkmaz, M. Krunz, “A Randomized Algorithm for Finding a Path Subject to Multiple QoS Constraints”, Computer Networks, vol. 36, pp. 251-268, 2001.

[19] T. Korkmaz, M. Krunz, “Multi-Constrained Optimal Path selection”, in Proceedings of IEEE INFOCOM 2001.

[20] X. yuan, “Heuristic Algorithms for Multiconstrained Quality-of-Service Routing”, IEEE/ACM Transactions on Networking, vol. 10, no. 2, April 2002.

[21] G. Liu, K.G. Ramakrishnan, “A*Prune: An Algorithm for Finding K Shortest Paths Subject to Multiple Constraints”, in Proceedings of IEEE INFOCOM 2001.

[22] P. Van Mieghem, F.A. Kuipers, T. Korkmaz, M. Krunz, M. Curado, E. Monteiro, X. Masip-Bruin, J. Solé-Pareta, S. Sànchez-López, “Quality of Service Routing”, Chapter 2 of book “QUALITY OF FUTURE INTERNET SERVICES: COST 263 FINAL REPORT”, Ed. Springer-Verlag, 2003, ISSN: 0302-9743 (in press), October 2003.

[23] R. Guerin, A. Orda, D. Williams, “QoS Routing Mechanisms and OSPF Extensions”, in Proceedings of 2nd Global Internet Miniconference (joint with Globecom'97), Phoenix, AZ, November 1997.

[24] Z. Wang, J. Crowcroft, “Quality-of-Service Routing for Supporting Multimedia Applications”, IEEE JSAC, 14(7): 1288-1234, September 1996.

[25] Y. Yang, L. Zhang, J.K. Muppala, S.T. Chanson, “Bandwidth-Delay Constrained Routing Algorithms”, Computer Networks, 2002.

[26] D.S. Reeves, H.F. Salama, “A distributed algorithm for delay-constrained unicast routing”, IEEE/ACM Transactions on Networking, 8(2):239-250, April 2000.

[27] A. Juttner, B. Szviatovszki, I. Mecs, Z. Rajko, “Lagrange relaxation based method for the QoS routing problem”, in Proceedings of IEEE INFOCOM 2001, vol. 2, pp. 859-868, April 2001.

[28] L. Guo, I. Matta, “Search space reduction in QoS routing”, in Proceedings of 19th III int. Conference on Distributed Computing Systems, III, May 1999, pp. 142-149.

[29] Q.Sun, H.Langendorfer, "A new distributed routing algorithm with end-to-end delay guarantee", in Proceedings of IWQoS'97, May 1997.
[30] A. Orda, "Routing with End-to-End QoS Guarantees in Broadband networks", IEEE/ACM Transactions on networking, vol. 7, no. 3, pp. 365-374, 1999.
[31] A. Orda, A. Sprintson, "QoS routing: the precomputation perspective", in Proceedings of IEEE INFOCOM 2000, pp. 128-136, 2000.
[32] R. Guerin, A. Orda, "Networks with advance reservations: The routing perspective", in Proceedings of IEEE INFOCOM 2000, Israel, March 26-30, 2000.
[33] N. Taft-Plotkin, B. Bellur, R. Ogier, "Quality-of-Service routing using maximally disjoint paths", in Proceedings of 7th International Workshop on Quality of service (IWQoS'99), London, England, pp. 119-128, May/June 1999.
[34] M. Kodialam, T.V. Lakshman, "Dynamic routing of bandwidth guaranteed tunnels with restoration", in Proceedings of IEEE INFOCOM 2000, pp. 902-911, 2000.
[35] M. Kodialam, T.V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering", in Proceedings of INFOCOM 2000.
[36] S.Suri, M.Waldvogel, P.R.Warkhede, "Profile-Based Routing: A New Framework for MPLS Traffic Engineering", in Proceedings of QofIS 2001, Coimbra, Portugal, September 2001.
[37] S. Vutukury, J.J. García-Luna Aceves, "A Simple Approximation to Minimum-Delay Routing", in Proceedings of ACM SIGCOMM'99, 1999.
[38] A. Khanna, J. Zinky, "The Revised ARPANET Routing Metric", in Proceedings of SIGCOMM'89, 1989.
[39] Z. Wang, J. Crowcroft, "Shortest Path First with Emergency Exits", in Proceedings of SIGCOMM'90, Philadelphia, USA, September 1990.
[40] K. Nahrstedt, S. Chen, "Coexistence of QoS and Best Effort Flows – Routing and Scheduling", in Proceedings of 10th IEEE Tyrrhenian International Workshop on Digital Communications: Multimedia Communications, Ischia, Italy, September 1998.
[41] Q. Ma, P. Steenkiste, "Supporting Dynamic Inter-Class Resource Sharing: A Multi-Class QoS Routing Algorithm", in Proceedings of IEEE INFOCOM 1999.
[42] A. Shaikh, J. Rexford, K. Shin, "Load-Sensitive Routing of Long-Lived IP Flows", in Proceedings of ACM SIGCOMM'99, 1999.
[43] J. Wang, K. Nahrstedt, "Hop-by-hop Routing Algorithms for Premium-class Traffic in Diffserv Networks", in Proceedings of IEEE INFOCOM 2002.
[44] M. Oliveira, J. Brito, B. Melo, G. Cuadros, E. Monteiro, "Quality of Service Routing in the Differentiated Services Framework", in Proceedings of SPIE's International Symposium on Voice, Video and Data Communications (Internet III: Quality of Service and Future Directions), Boston, Massachusetts, USA, November 5-8, 2000.
[45] M. Curado, O. Reis, J. Brito, G. Cuadros, E. Monteiro, "Stability and Scalability Sigues in Hop-by-Hop Class-based Routing", in Proceedings of 2nd International Workshop on QoS in Multiservice IP Networks, QoS-IP2003, Milano, Italy, February 24-26, 2003.
[46] ATM Forum, af-pnni-0055.002, "Private Network to Network Interface Specification Version 1.1", April 2002.
[47] G.Apostopoulos, R.Guerin, S.Kamat and S.Tripathi, "Quality of Service based routing: A performance perspective", in Proceedings of SIGCOMM, Vancouver, September 1998.
[48] B.Lekovic and P.Van Mieghem, "Link State Update Policies for Quality of service Routing", in Proceedings of 8th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT2001), Delft, The Netherlands, October 18, pp. 123-128.
[49] G.Apostopoulos, R.Guerin, S.Kamat, S.K.Tripathi, "Improving QoS Routing Performance Under Inaccurate Link State Information", in Proceedings of ITC'16, June 1999.
[50] Guerin, R., Orda, A., "QoS Routing in Networks with Inaccurate Information: Theory and Algorithms", IEEE/ACM Transactions on Networking, vol.7, nº:3, pp.350-364, June 1999.
[51] D.H. Lorenz, A.Orda, "QoS Routing in Networks with Uncertain Parameters", IEEE/ACM Transactions on Networking, vol.6, nº:6, December 1998.
[52] S.Chen, K.Nahrstedt, "Distributed QoS Routing with Imprecise State Information", in Proceedings of 7th IEEE International Conference of Computer, Communications and Networks, Lafayette, LA, pp614-621, October 1998.
[53] S.Kim, M.Lee, "Server Based QoS Routing with Implicit network State Updates", in Proceedings of IEEE Globecom 2001, vol.4, pp.2182-2187, San Antonio, Texas, November 2001.

[54] S.Nelakuditi, Z.Zhang, R.P.Tsang, "Adaptive Proportional Routing: A localized QoS Routing Approach", in Proceedings of INFOCOM 2000, pp.1566-1575.
[55] B.Awerbuch, B.Berger, L.Cowen, D.Peleg, "Fast network decomposition", in Proceedings of 11th Ann. Symp. Principles of Distributed Computing", pp. 169-177, Vancouver, BC, Canada, August 1992.
[56] B.Awerbuch, B.Berger, L.Cowen, D.Peleg, "Near lineal cost sequential and distributed constructions of sparse neighborhood covers", in Proceedings of 34th Annu Symp, Foundations of Computer Science, Palo Alto, November 1993
[57] N.Linial, M.Saks, "Low diameter graph decompositions", Combinatorica, vol.13, n0.4, pp.441-454. 1993.
[58] Y.Bartal, "Probabilistic approximation of metric spaces and its algorithm applications", in Proceedings of 37th Annu. Symp.Foundations of Computer science, pp.184-193, Burlington, VT, October 1996.
[59] B.Awerbuch, B.Berger, L.Cowen, D.Peleg, "Fast distributed network decomposition and covers", J.Parallel Distrib.Comput., vol.39, no 2, pp.105-114, December 1996.
[60] M.R.Garey, D.S.Johnson, "Computers and Intractability", San Francisco, CA: Freeman, 1979.
[61] G.Apostopoulos, R.Guerin, S.Kamat, S.K.Tripathi, "Server Based QoS Routing", in Proceedings of Global Communications Conference IEEE Globecom 1999.
[62] X.Masip-Bruin, S.Sánchez-López, J.Solé-Pareta, J.Domingo-Pascual, "A QoS Routing Mechanism for Reducing the Routing Inaccuracy Effects", in Proceedings of QoS-IP 2003, pp.90-102, Milan, Italy, Feb. 2003.
[63] T.M.Chen, T.H.Oh, "Reliable Services in MPLS", IEEE Communications Magazine, 1999, pp. 58-62.
[64] X.Masip-Bruin, S.Sánchez-López, J.Solé-Pareta, J.Domingo-Pascual, "An Alternative Path Fast Rerouting in MPLS", in Proceedings of ISCIS XV, Istanbul, Turkey, October 2000.
[65] Q. Ma, P. Steenkiste, "On Path Selection for Traffic with Bandwidth Guarantees", in Proceedings of IEEE International Conference on Networks Protocols, October 1997
[66] X.Masip-Bruin, S.Sánchez-López, J.Solé-Pareta, J.Domingo-Pascual, "QoS Routing Algorithms under Inaccurate Routing Information for Bandwidth Constrained Applications", in Proceedings of International Communications Conference, IEEE ICC'03, Anchorage, Alaska, May 2003
[67] X. Masip-Bruin, R. Muñoz, S. Sànchez-López, J. Solé-Pareta, J. Domingo-Pascual, G. Junyent, "An Adaptive Routing Mechanism for Reducing the Routing Inaccuracy Effects in an ASON", in Proceedings of 7th IFIP Working Conference on Optical Network Design & Modelling, pp. 333-349, Budapest, Hungary, February 2003.
[68] X. Masip-Bruin, S. Sànchez-López, J. Solé-Pareta, J. Domingo-Pascual, D. Colle, "Routing and Wavelength Assignment under Inaccurate Routing Information in Networks with Sparse and Limited Wavelength Conversion", accepted for publication in Global Communications Conference, IEEE GLOBECOM'03, San Francisco, USA, December 2003.
[69] S.Sánchez-López, X. Masip-Bruin, J.Solé-Pareta, J.Domingo-Pascual, "PONNI: A Routing Information Exchange Protocol for ASON", in Proceedings of Eurescom Summit 2002, Germany, Oct.2002
[70] H.Zang, J.P.Jue, B.Mukherjee, "A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks", Optical Networks Magazine, January 2000.
[71] K.Chan, T.P.Yun, "Analysis of Least Congested Path Routing in WDM Lightwave Networks", in Proceedings of INFOCOM'94, vol.2, pp.962-969.
[72] Y. Mei, Ch. Qiao, "Efficient distributed control protocols for WDM all-optical networks", in Proceedings of IC3N'97, Las Vegas, NV, pp. 150-153, September 1997.
[73] J. Solé-Pareta, X. Masip-Bruin, S. Sànchez-López, S. Spadaro, D. Careglio, "Some Open Issues in the Optical Networks Control Plane", in Proceedings of International Conference on Transparent Optical networks, IEEE ICTON 2003, Warsaw, Poland, June 2003.
[74] J.Zhou, X.Yuan, "A Study of Dynamic Routing and Wavelength Assignment with Imprecise Network State Information", in Proceedings of ICPP Workshop on Optical Networks, Canada, August 2002
[75] J.Zheng, H.T.Mouftah, "Distributed lightpath control based on destination routing in wavelength-routed WDM networks", Optical Networks Magazine, Vol.3, nº:4, pp.38-46, July/August 2002.

[76] S. Darisala, A. Fumagalli, P. Kothandaraman, M. Tacca, L. Valcarenghi, "On the Convergence of the Link-State Advertisement Protocol in Survivable WDM Mesh Networks", in Proceedings of 7th IFIP Working Conference on Optical Network Design & Modelling, ONDM'03, pp. 433-447, Budapest, Hungary, February 2003.

[77] S. Ramamurthy, B. Mukherjee, "Survivable WDM Mesh Networks. Part I-Protection", in Proceedings of IEEE INFOCOM'99, vol. 2, pp. 744-751, 1999.

[78] F.Forghieri, A. Bononi, P.R.Prucnal, " Analysis and Comparison of Hop-Potato and Single-Buffer Deflection Routing in Very High Bit rate Optical Mesh Networks", IEEE Transactions on Communications, vol. 43, no. 1, pp. 88-98, January 1995.

[79] J.P.Jue, G.Xiao, "An Adaptive Routing Algorithm for Wavelength-Routed Optical Networks with a Distributed Control Scheme", in Proceedings of 9th Int'l.Conf.Comp.Commun.Net.Las Vegas, NV, pp.192-197, Oct.2000.

[80] J.Yates, J.Lacey, D.Everitt, M.Summerfield, "Limited-range Wavelength Translation in All-Optical Networks", in Proceedings of IEEE INFOCOM 1996, pp.954-961.

APPENDIX A:

List of Publications and Projects

Main publications

1) X.Masip-Bruin, S.Sànchez-López, J.Solé-Pareta, J.Domingo-Pascual,, "*An Alternative Path Fast Rerouting in MPLS*", in Proceedings of ISCIS XV, pp.304-313, Istambul, Turkey, October 2000.

2) X.Masip-Bruin, R.Muñoz, S.Sànchez-López, J.Solé-Pareta, J.Domingo-Pascual, G.Junyent, "*Mecanismo de Encaminamiento Dinámico en Redes ASON*", XII Jornadas Telecom I+D, Barcelona, Spain, November 2002.

3) X.Masip-Bruin, R.Muñoz, S.Sànchez-López, J.Solé-Pareta, J.Domingo-Pascual, G.Junyent, "*An Adaptive Routing Mechanism for Reducing the Routing Inaccuracy Effects in an ASON*", in Proceedings of 7th IFIP Working Conference on Optical Network Design & Modelling, pp. 333-349, Budapest, Hungary, Feb. 2003.

4) X.Masip-Bruin, S.Sànchez-López, J.Solé-Pareta, J.Domingo-Pascual,, "*A QoS Routing Mechanism for Reducing the Routing Inaccuracy Effects*", in Proceedings of 2nd International Workshop on QoS in Multiservice IP Networks, QoS-IP, pp.90-102, Milan, Italy, February 2003.

5) X.Masip-Bruin, S.Sànchez-López, J.Solé-Pareta, J.Domingo-Pascual, "*A Mechanism to Reduce the Routing Information Inaccuracy Effects: Application to MPLS"*, Workshop on MPLS, Girona, March 2003.

6) X.Masip-Bruin, S.Sànchez-López, J.Solé-Pareta, J.Domingo-Pascual,, "*QoS Routing Algorithms Under Inaccurate Routing Information for Bandwidth Constrained Applications*", in Proceedings of IEEE International Communications Conference, ICC 2003, Anchorage, Alaska, May 2003.

7) J.Solé-Pareta, D.Careglio, X.Masip-Bruin, S.Sanchez-López and S.Spadaro, "*Some Open Issues in the Definition of a Control Plane for Optical Networks*", Invited paper submitted to IEEE 5th International Conference on Transparent Optical Networks, ICTON 2003, Warsaw, Poland, June 2003.

8) S. De Maesschalck, D. Colle, B. Puype, Q. Yan, M. Pickavet, P. Demeester, S. Sánchez-López, X. Masip-Bruin, J. Solé-Pareta, et al, "*Circuit/Wavelength Switching and Routing*", in Proceedings of 7th. International Conference on Telecommunications, ConTEL 2003, Zagreb, Croatia, June, 2003.

9) P. Van Mieghem, F.A. Kuipers, T. Korkmaz, M. Krunz, M. Curado, E. Monteiro, X. Masip-Bruin, J. Solé-Pareta, S. Sànchez-López, "*Quality of Service Routing*", Chapter 2 of book "QUALITY OF FUTURE INTERNET SERVICES: COST 263 FINAL REPORT", Ed. Springer-Verlag, 2003, ISSN: 0302-9743 (in press), October 2003

10) X. Masip-Bruin, S. Sànchez-López, J. Solé-Pareta, et al. Contribution to Chapter 3 on "*Design optimized reliable WDM networks*" of COST 266 Final Report, October, 2003.

11) X. Masip-Bruin, S. Sànchez-López, J. Solé-Pareta, J. Domingo-Pascual, D. Colle, "*Routing and Wavelength Assignment under Inaccurate Routing Information in Networks with Sparse and Limited Wavelength Conversions"*, in Proceedings of Global Communications Conference, IEEE GLOBECOM 2003, San Francisco, CA, December 2003.

12) X. Masip-Bruin, S. Sànchez-López, J. Solé-Pareta, J. Domingo-Pascual, "*Hierarchical Routing with QoS Constraints in Optical Transport Networks*", submitted to IEEE INFOCOM 2004.

Other publications

1) S.Sànchez-López, X.Masip-Bruin, J.Domingo-Pascual, "*Protocolo RSVP: evolución y experiencias*", Jornadas Técnicas RedIRIS 98, Noviembre 1998, Barcelona, Spain.
2) S.Sànchez-López, X.Masip-Bruin, J. Domingo-Pascual, J. Solé-Pareta, "*A Solution for Integrating MPLS over ATM*", in Proceedings of ISCIS XV, Istambul, Turkey, October 2000.
3) S.Sànchez-López, X.Masip-Bruin, J.Domingo-Pascual, J.Solé-Pareta, J.López, "*A Path Establishment Approach in an MPLS-ATM Integrated Environment*", in Proceedings of IEEE GLOBECOM, S.Antonio, Texas, Nov. 2001.
4) S.Sànchez-López, X.Masip-Bruin, J.Solé-Pareta, J.Domingo-Pascual, "*Providing QoS in MPLS-ATM Integrated Environments*", in Proceedings of QofIS'02, Zürich, Switzerland, October 2002.
5) S.Sànchez-López, X.Masip-Bruin, J.Solé-Pareta, J.Domingo-Pascual, "*PONNI: A Routing Information Exchange Protocol for ASON*", in Proceedings of Eurescom Summit 2002, Heidelberg, Germany, October 2002.

Projects

This work has been supported by the following projects: the *Spanish Ministry of Education* (*CICyT*) under contracts TIC99-0572-C02-02 and TEL99-1117-C03-3; the *Spanish Ministry of Science and Technology* (*MCyT*) under contracts FEDER-TIC2002-04531-C04-02 and FEDER-TIC2002-04344-C02-02; the *Catalan Research Council* (*CIRIT*) under contracts 1999-SGR00126 i 2001-SGR00226; and the COST 263 Action on "Quality of Future Internet Services".

APPENDIX B:

The ns/2 Simulator

The ns/2 simulator is a discrete event simulator targeted at networking research. The ns/2 provides substantial support for simulation of *TCP*, routing and multicast protocols over wired and wireless networks. Even though it began as a variant of the *REAL network simulator* in 1989, it is not a polished and finished product, but the result of an on-going effort of research and development. In fact, the ns/2 is still being improved and modified by most users around the world. Currently, there are several web pages where information about bugs, updated versions and comments can be found.

The ns/2 was the first option when locking for a network simulator where network performance could be evaluated. There are two main reasons supporting this decision: it is free (that is a good reason), and it provides users with completely freedom to modify any network parameter, from network topology to traffic characteristics and routing mechanisms. Therefore, the ns/2 is very attractive because it allows us to include in the network mechanisms developed in this Thesis and to evaluate their performance.

The ns/2 simulator has been properly modified to support *MPLS* capabilities, to implement new routing mechanisms and finally to represent a *WDM* network.

Concerning the *MPLS* module only required features have been developed. Basically these are the following:

- Label generation
- Label distribution
- Explicit LSP set-up
- Label space
- Label control
- Signalling mechanisms
- Label Distribution Protocol
- Label allocation

As far as the routing mechanisms concerns, all the routing algorithms inferred from the *BBR* mechanism have been implemented. Therefore, the following routing algorithms are included:

- *SP*: Shortest Path
- *WSP*: Widest-Shortest Path
- *SSP*: Shortest-Safest Path
- *SOSP*: Shortest-Obstruct-Sensitive Path
- *OSSP*: Obstruct-Sensitive-Shortest Path
- *WSOSP*: Widest-Shortest-Obstruct-Sensitive Path
- *BOSP*: Balance-Obstruct-Sensitive Path
- *BOSP/BDP*: *BOSP* along with the *BYPASS Discovery Process*
  - Limiting by a fixed value the number of computed *bypass-paths* per route
  - Limiting the number of computed *bypass-paths* depending on the network load

Finally, the simulator has been extended with optical capabilities. This really means that some features of *WDM* networks may be simulated when including our modifications. Basically, the simulator has been provided with the following:

- Selects more than one feasible shortest route
- Modifies the network state databases and the update messages to include optical information, such as number of fibres, number of wavelengths, and wavelength availability
- Includes wavelength conversion capabilities. Sparse and limited wavelength conversion is allowed.

The *BBOR* mechanism is developed in this optical network scenario, therefore the following routing algorithms are implemented in the simulator:

- SP + ALG1: Algorithm 1
- SP + ALG2: Algorithm2
- SP + ALG3: Algorithm 3
- SP + First-Fit: First-Fit