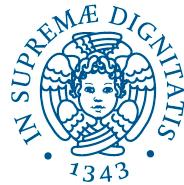


UNIVERSITÀ DI PISA

Scuola di Dottorato in Ingegneria "Leonardo da Vinci"



**Corso di Dottorato di Ricerca in
INGEGNERIA DELL'INFORMAZIONE**

Tesi di Dottorato di Ricerca

Intelligent Network Infrastructures: New Functional Perspectives on Leveraging Future Internet Services

Autore:

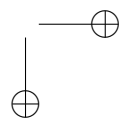
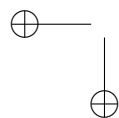
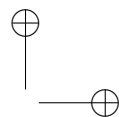
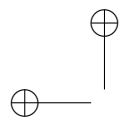
Luiz Gustavo Zuliani _____

Relatori:

Prof. Stefano Giordano _____

Prof. Franco Russo _____

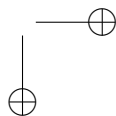
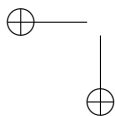
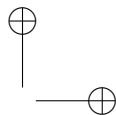
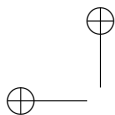
*Anno 2011
SSD ING-INF/03*



Ai miei genitori

Aos meus pais

Domingos e Maria Lúcia



Sommario

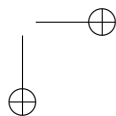
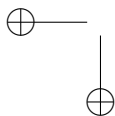
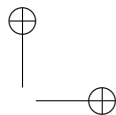
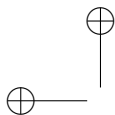
La rete Internet del 21° secolo è molto diversa da quella dei primi anni 80. Essa nel tempo si è adattata, diventando una piattaforma di business di grande successo su scala globale. Come ogni tecnologia di successo, ha subito un processo naturale di ossificazione. Negli ultimi 30 anni, per poter far fronte a nuove applicazioni emergenti si è cercato di aggiungere nuove funzionalità, estendendo i protocolli esistenti, creando reti overlay, oppure ricorrendo all'utilizzo di collegamenti a banda sempre più elevata. Purtroppo, questo approccio non è idoneo per un'ampia gamma di nuove applicazioni che per poter essere utilizzate richiedono alla rete funzionalità così avanzate che lo stack TCP/IP e i suoi derivati non possono fornire. A tal proposito, le reti della prossima generazione (*next generation networks*, oppure semplicemente NGN) sono pensate proprio per supportare i futuri servizi Internet. Questa tesi contribuisce con tre proposte a questo ambizioso obiettivo.

La prima proposta presenta un'architettura preliminare che permette alle NGN di richiedere in modo trasparente servizi avanzati a livello 1, come per esempio la QoS e i circuiti punto-multipunto. Questa architettura è basata su tecniche di virtualizzazione applicate al livello 1 che mascherano alle NGN tutte le complessità che riguardano la fornitura di circuiti inter-dominio. Sono stati considerati anche gli aspetti economici, rendendo l'architettura appetibile ai carrier.

Il secondo contributo riguarda un framework per lo sviluppo di una rete DiffServ-MPLS basata esclusivamente su software open source e comuni PC. Inoltre, un software router DiffServ-MPLS è stato progettato per consentire la prototipazione di NGN che, come elemento iniziale di sviluppo, fanno uso di pseudo circuiti virtuali e qualità del servizio garantita.

Infine, si propongono algoritmi per il routing e l'assegnamento delle lunghezze d'onda (*routing and wavelength assignment*) nelle reti fotoniche. Tali algoritmi tengono conto delle restrizioni a livello fisico per garantire al 100% il profilo di QoS richiesto anche in caso di un guasto nella rete. Nuove tecniche sono state introdotte in modo da garantire una probabilità di blocco minore rispetto agli algoritmi che rappresentano lo stato dell'arte, senza tuttavia peggiorare il tempo di setup.

Parole chiave: servizi dell'Internet del futuro, reti di prossima generazione, infrastrutture di rete intelligente, virtualizzazione, reti fotoniche.



Abstract

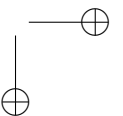
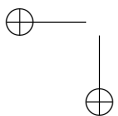
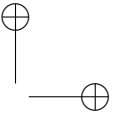
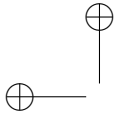
The Internet experience of the 21st century is by far very different from that of the early '80s. The Internet has adapted itself to become what it really is today, a very successful business platform of global scale. As every highly successful technology, the Internet has suffered from a natural process of ossification. Over the last 30 years, the technical solutions adopted to leverage emerging applications can be divided in two categories. First, the addition of new functionalities either patching existing protocols or adding new upper layers. Second, accommodating traffic grow with higher bandwidth links. Unfortunately, this approach is not suitable to provide the proper ground for a wide gamma of new applications. To be deployed, these future Internet applications require from the network layer advanced capabilities that the TCP/IP stack and its derived protocols can not provide by design in a robust, scalable fashion. NGNs (*Next Generation Networks*) on top of intelligent telecommunication infrastructures are being envisioned to support future Internet Services. This thesis contributes with three proposals to achieve this ambitious goal.

The first proposal presents a preliminary architecture to allow NGNs to seamlessly request advanced services from layer 1 transport networks, such as QoS guaranteed point-to-multipoint circuits. This architecture is based on virtualization techniques applied to layer 1 networks, and hides from NGNs all complexities of interdomain provisioning. Moreover, the economic aspects involved were also considered, making the architecture attractive to carriers.

The second contribution regards a framework to develop DiffServ-MPLS capable networks based exclusively on open source software and commodity PCs. The developed DiffServ-MPLS flexible software router was designed to allow NGN prototyping, that make use of pseudo virtual circuits and assured QoS as a starting point of development.

The third proposal presents a state of the art routing and wavelength assignment algorithm for photonic networks. This algorithm considers physical layer impairments to 100% guarantee the requested QoS profile, even in case of single network failures. A number of novel techniques were applied to offer lower blocking probability when compared with recent proposed algorithms, without impacting on setup delay time.

Keywords: future Internet services, next generation networks, intelligent network infrastructures, virtualization, photonic networks.



Ringraziamenti

Vorrei esprimere la mia piena gratitudine:

Ai miei tutori Prof. Stefano Giordano e Prof. Franco Russo, per l'orientamento e le opportunità concessemi.

All'Ing. Davide Adami, con il quale ho avuto la fortuna di lavorare da vicino sin dall'inizio. Sempre disposto a darmi una mano, anche sugli argomenti al di là della ricerca. Senza i suoi numerosi contributi, questa tesi non sarebbe stata possibile. Più di un collega d'ufficio, un caro amico.

Al Prof. Michele Pagano e all'Ing. Rosario Garroppo, per i numerosi consigli, suggerimenti e aiuti vari.

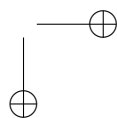
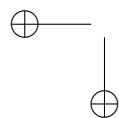
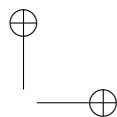
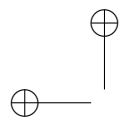
A tutti i miei colleghi di dipartimento, in particolare agli amici Gianni Antichi e Davide Iacono con i quali è stato un piacere lavorare nel corso di questi anni.

Ai miei tesisti Enrico Gloria, Gianluca Epifano, Luigi Li Calzi e Marcello Gollinucci. Spero che il nostro periodo di collaborazione sia stato per voi proficuo come lo è stato per me.

Ai dipendenti della Portineria e della Segreteria del dipartimento, in special modo a Simone Kovatz, Maura Pancanti, Donatella Giorgi e Rosana Le Rose: siete stati sempre pazienti e gentilissime, perfino quando le mie domande erano già troppe.

Alla mia famiglia, per il supporto illimitato e incondizionato.

Al Cav. Walter Ricciluca la cui serietà e integrità costituiscono un riferimento costante per me.



Contents

List of Figures	XIII
List of Tables	XV
Papers Published by the Author	XVII
1 Introduction	1
1.1 Motivations	5
1.2 Contributions	6
1.3 Text Organization	6
2 Background	7
2.1 Photonic Networks	7
2.1.1 Optical Fibers	8
2.1.2 Optical Transmitters and Receivers	8
2.1.3 Physical Impairments and Regenerators	9
2.1.4 Optical Amplifiers	10
2.1.5 WRPNs	10
2.1.6 RWA	12
2.2 Internet and QoS	14
2.2.1 IntServ	15
2.2.2 DiffServ	16
2.3 MPLS Switching	19
2.4 DS and MPLS integration	21
2.5 GMPLS Architecture	22
2.5.1 LMP	23
2.5.2 OSPF-TE	24
2.5.3 RSVP-TE	25
2.5.4 PCE and PCEP	26
2.6 Network Virtualization	27

3	Transport Network Virtual Environments	31
3.1	Related Works and TNVE presentation	31
3.2	Internet and TNVE Business models	33
3.3	Application Scenario	36
3.4	FIE Architecture	37
3.5	Case Study	44
4	Open Framework for Service Software Routers	49
4.1	Preamble	49
4.2	Related Works	50
4.3	Framework Architecture and Features	52
4.3.1	Data Plane	55
4.3.2	Control Plane	61
4.4	Open Framework Live Distributions	68
5	QoT-Assured Survivable LP Provisioning	73
5.1	State of the Art	73
5.1.1	Network Impairment Models	74
5.1.2	Resilience	75
5.1.3	Methods for solving the RWA problem	76
5.1.4	Performance Evaluation Metrics	76
5.2	Proposed Algorithms	77
5.2.1	MCP-RWA	77
5.2.2	MCP-D ²	89
5.2.3	MCP-S	89
6	Conclusion	99
	References	113
	List of Acronyms	115
A	Manual Configuration of a Software Router DP	121
B	TED Description	129
C	VNT Standard Topology Configuration	137

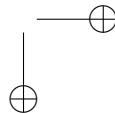
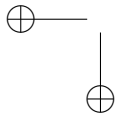
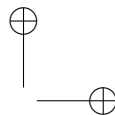
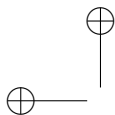
List of Figures

2.1	A PXC formed by M fibers, each one carrying K wavelengths. . . .	11
2.2	An IP network on top of a WRPN.	12
2.3	The TOS octet, part of the IP header.	14
2.4	The DS Field structure (redefinition of the TOS octet of the IP header).	17
2.5	Functional flowchart of the DiffServ Model.	18
2.6	Link layer and network protocols that can be integrated by the MPLS standard.	19
2.7	Shim header between the link layer and network headers.	20
2.8	A TLV triple.	25
3.1	Internet business model.	34
3.2	TNVE business model.	35
3.3	TNVE scenario.	38
3.4	FIE architecture.	40
3.5	A slice space.	40
3.6	P2MP circuit provisioning.	43
3.7	MP2P circuit provisioning.	44
3.8	Case study scenario.	45
4.1	DS-MPLS Open Framework architecture.	54
4.2	Hierarchical scheduler tree.	59
4.3	E-LSP scheduler subtree.	60
4.4	L-LSP scheduler subtrees for AF and EF traffics.	60
4.5	Collaboration diagram for the centralized LSP setup procedure.	63
4.6	Collaboration diagram for the centralized LSP teardown procedure.	64
4.7	LCS internal components and the PCE interaction.	65
4.8	LSP setup request sequence diagrams.	66
4.9	LSP teardown request sequence diagrams.	67
4.10	Packet trace containing PATH and RESV messages.	68
4.11	One of the DS-MPLS live distribution boot splash screen.	69
4.12	TNV topology.	70

5.1	MCP-RWA macro-level flowchart.	78
5.2	Cascading of amplifiers.	82
5.3	MCP-RWA detailed flowchart.	83
5.4	Cost function comparison.	85
5.5	Criticality threshold comparison.	86
5.6	MCP-RWA mean blocking probability.	87
5.7	Confidence intervals for MCP-RWA mean blocking probability. . . .	88
5.8	MCP-RWA processing time.	88
5.9	MCP-D ² detailed flowchart.	90
5.10	MCP-S detailed flowchart.	91
5.11	Survivable IA-RWA grid topology blocking probability.	93
5.12	Survivable IA-RWA NSFNET topology blocking probability.	93
5.13	Survivable IA-RWA Italian topology blocking probability.	94
5.14	Survivable IA-RWA American topology blocking probability.	94
5.15	Survivable IA-RWA grid topology processing time.	95
5.16	Survivable IA-RWA NSFNET topology processing time.	95
5.17	Survivable IA-RWA Italian topology processing time.	96
5.18	Survivable IA-RWA American topology processing time.	96

List of Tables

2.1	DSCP values for all AF X classes and Y drop precedences.	18
3.1	Association between NGNEs and FIEs.	45
3.2	TNA Members Characteristics.	46
3.3	Interconnection Links TE Properties.	46
4.1	Association between DSCP and EXP fields used to setup E-LSPs .	56
4.2	Example of association between FECs and LSPs	57



Papers Published by the Author[†]

Journal articles

1. D. Adami, S. Giordano, M. Pagano, L. G. Zuliani, “Multidomain Layer 1 Infrastructure Virtualization as a Future Internet Services-enabling Paradigm,” *Journal of Internet Engineering (JIE)*, Vol. 4, N. 1, pp. 251–259, Dec. 2010.
2. G. S. Pavani, L. G. Zuliani, H. Waldman, M. Magalhães, “Distributed Approaches for Impairment-aware Routing and Wavelength Assignment algorithms in GMPLS Networks,” *Computer Networks (COMNET)*, Vol. 52, N. 10, pp. 1905–1915, July 2008.

Conference proceedings papers

3. D. Adami, S. Giordano, M. Pagano, L. G. Zuliani, “Online Lightpath Provisioning and Critical Services: A new IA-RWA algorithm to Assure QoT and Survivability,” *IEEE International Conference on High Performance Switching and Routing (HPSR 2011)*, Cartagena, Spain, to appear.
4. D. Adami, S. Giordano, M. Pagano, L. G. Zuliani, “MCP-RWA: A Novel Algorithm for QoT-guaranteed Online Provisioning in Photonic Networks,” *International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT 2010)*, Moscow, Russia, Oct. 2010.
5. D. Adami, S. Giordano, M. Pagano, L. G. Zuliani, “Lightpath Survivability with QoT Guarantees: Developing and Evaluating a New Algorithm,” *Int. Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2010)*, Ottawa, Canada, July 2010.
6. D. Adami, S. Giordano, M. Pagano, L. G. Zuliani, “Lightpath Survivability with QoT Guarantees: Key Issues in Wavelength-Routed Photonic Networks,” *Int. Conference on Optical Communication Systems (OPTICS 2010)*, Athens, Greece, July 2010.
7. D. Adami, S. Giordano, M. Pagano, L. G. Zuliani, “On Leveraging Future Internet Services Through Multidomain Layer 1 Virtualization,” *IEEE 3rd Workshop on Enabling the Future Service-Oriented Internet (EFSOI’09)*, Honolulu, USA, Dec. 2009.
8. D. Adami, S. Giordano, M. Pagano, L. G. Zuliani, “A Flexible Software Router based Framework to Enable a DS-MPLS Transport Network,” *IEEE 14th Int. Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD’09)*, Pisa, Italy, June 2009.

[†]The list of authors of all papers exclusively written by TLCNETGRP members are sorted by surname. The author of this thesis acknowledges to be the corresponding author of paper n. 1 and papers n. 3 through 9.

9. D. Adami, S. Giordano, F. Russo, L. G. Zuliani, “A New Framework for the Development and Deployment of Multi-purpose Service Specific Overlay Networks,” *Italian Networking Workshop*, Cortina d’Ampezzo, Italy, Jan. 2009.

Previously published papers

10. L. G. Zuliani, M. Savasini, G. S. Pavani, R. Pasquini, F. L. Verdi, and M. Magalhães, “An implementation of an OSPF-TE to support GMPLS-controlled All-Optical WDM Networks,” *VI International Telecommunications Symposium (ITS2006)*, Fortaleza, Brazil, Sept. 2006.
11. R. Pasquini, F. L. Verdi, L. G. Zuliani, M. Magalhães, and S. Rossi, “An Optical UNI Architecture for the GIGA Project Testbed Network,” *VI International Telecommunications Symposium (ITS2006)*, Fortaleza, Brazil, Sept. 2006.

Chapter 1

Introduction

Since its creation, the Internet has evolved from an experimental, packet-switched research network to a public, global telecommunication infrastructure that supports applications in all areas of knowledge. As well known, the Internet is a “network of networks”, i.e. a conglomerate of networks mainly owned by large corporations, government and research institutions, universities and ISPs (*Internet Service Providers*). These administrative domains operate their networks in an independent way, using specific technologies and customized settings (ranging from transport devices to routing and access policies) to best fit their purposes. Nowadays, Internet is composed of thousands of such interconnected networks that are organized as ASes (*Autonomous Systems*). As described in [1], the IANA (*Internet Assigned Numbers Authority*) has begun to assign 32 bits AS numbers, due to the near exhaustion of the old 16 bits format pool. Interoperability among ASes is still based on the original TCP/IP protocol architecture, which was conceived to deliver data in a connectionless, best-effort fashion. Therefore, globally offered Internet services must rely exclusively on the TCP/IP stack of protocols. The main concern by the time of TCP/IP creation was to effectively provide reachability between applications running on different hosts. By the early '80s, TCP/IP was successful on supporting the applications of that time, namely rsh, telnet, ftp and e-mail.

From the beginning, Internet itself can be seen as a data overlay network on top of a telecommunication infrastructure. Leased lines were deployed to interconnect networks in metro and long-haul (cross-country and overseas) areas, using data-link control protocols like HDLC (*High-level Data Link Control*). Later on, access was also provided using POTS (*Plain Old Telephone Service*) and SLIP (*Serial Line Internet Protocol*), among others. As in any overlay, the upper layer benefits from the under layer resources in transparent or quasi-transparent manner. The Internet overlay was - and still is - no exception to this rule: IP routers have little or no information at all regarding the number and the operational status

of the L1 (*Layer 1*) circuits that compose interconnection IP links. This was a design feature: TCP/IP would deal with malformed and missing packets, while applications would just have to use the (newly available) sockets API (*Application Programming Interface*) to reliably transmit information across two end hosts.

Internet applications have changed drastically in the last fifteen years. On-demand multimedia delivery, content exchange, videoconferencing, massive multiplayer entertainment, social networks and virtual realities are now part of our daily lives. Two main factors have made possible the emergence of such applications: the tremendous increase of CPU power and network capacity at affordable prices, and the huge effort put by the academic and industry communities to add to the Internet the network functionalities requested by today's applications. The latter was realized using the so called *incremental approach*, that can be summarized with the development of new IP-based protocols and extensions of the existing ones, and also with the creation of new overlay solutions on top of the Internet. As the IP network layer has no visibility of its underlayers, i.e. the telecommunication transport infrastructure, solutions have to rely on simple monitoring tools (like ping or traceroute) and/or complex probabilistic algorithms in order to minimize service disruption and maximize the QoE (*Quality of Experience*) of users. Therefore, the Internet as it is today can globally offer, under the best conditions, just better-than-best-effort services [2, 3].

Considering the same period of the last fifteen years, a wide range of network challenges were overcome with innovative solutions, specially in the field of seamless mobile and QoS-guaranteed (*Quality of Service*) communications. These advances, however, have seen little to none deployment on the Internet at large. Unfortunately, the Internet has suffered from a natural evolutionary process that any successful technology is subjected to. This stagnation process is called *ossification*, and poses a severe obstacle to the continuing evolution (and even the survival) of the Internet [4]. The ICANN (*Internet Corporation for Assigned Names and Numbers*) made an official communication on February 3rd, 2011 stating that the IPv4 address pool is already depleted [5]. By the time of this writing, it is predicted that all IPv4 address still not allocated by ICANN-affiliated regional organizations will be over in the next months. Even facing this critical situation, the global deployment of IPv6 [6] (the next version of IP, that uses 128 bits of addressing space instead 32 bits of IPv4) is still not a reality. So far, incremental approach solutions were convenient. In fact, the potential service disruption caused by the adoption of new, non IP-compatible protocols could lead to catastrophic financial consequences, to say the least.

A whole myriad of innovative, highly profitable services is being predicted for the future Internet, driven by a number of factors such as quality and cheap ubiquitous access (anywhere, at any time, and from any device), the proliferation of user-generated content, and the new advances in human-computer interaction. These new services will require new (or higher levels of) capabilities concerning

seamless access, mobility and multi-homing, content- and context-awareness, privacy, anonymity, security, identity and trust management, accounting, and QoS [7]. These future Internet services simply can not be leveraged by just adding new functionalities to end systems, leaving the network core untouched. Moreover, many (if not all) network functionalities essential to future Internet services can not be achieved by IP-based protocols, at least without facing scalability issues [8]. Between all of these functional requirements, one of the most delicate is Internet QoS. IP QoS is complex, expensive and not straight-forward to implement even in small scale scenarios. It is much simpler and cheaper to workaround QoS-related problems with overprovisioning. As stated in [3], “... trying to introduce QoS in IP routed and connectionless networks is indeed a Utopian idea. This is like trying to introduce fine cuisine, gourmet dishes, and a la carte menus in fast food restaurants. In other words, IP QoS is not the right approach for Internet QoS.”.

To circumvent current Internet limitations and to provide the proper ground to leverage advanced future Internet services, new network architectures and protocols suites are being envisioned. Proposals for networks that incorporate such pioneer elements are commonly called NGNs (*Next Generation Networks*). It is worth to note that the term NGN in this thesis do not intend to make reference to any specific network solution, unless when explicitly cited. So far, the available NGN proposals are very heterogeneous (ranging from sub-IP protocols to complete clean slate designs [9]), and usually are data transmission technology independent. It is still unclear how NGNs will be realized, but a number of paradigm-shifts already have been defined. While current Internet is based on best-effort service provisioning, NGNs will feature assured provisioning. Instead of providing communication capabilities between end-hosts, connectivity services will be provided by NGNs upon application request. On-request service invocation will be replaced by on-demand service invocation. Finally, the vertically-oriented, application-specific approach will give place to a horizontally-integrated, multi-service network [10].

There appear to be a consensus that, while focusing on user services, NGNs should keep the interaction with the underlay infrastructure as transparent as possible. Indeed, the wide variety of data hauling technologies and the intricacy of their organizational structure would pose a huge obstacle to the development of NGNs, if managed with full visibility. Some questions arise from this observation: will it be possible to meet future applications requirements - specially QoS related ones - without visibility of the underlay infrastructure? If so, how could it be achieved? The answer could rely on network virtualization. By this principle, multiple, specialized NGNs could share a single physical substrate, without compromising flexibility, manageability, fault tolerance, security and privacy [11]. In the future Internet scenario, the physical substrate is the telecommunication infrastructure that provides backbone and long haul L1 optical circuits. It is expected that this infrastructure will be primary composed by WRPNS

(*Wavelength Routed Photonic Networks*) controlled by the GMPLS (*Generalized MultiProtocol Label Switching*) [12] suite of protocols [13]. Therefore, the control entities of the virtualized physical substrate must orchestrate multiple GMPLS L1 networks in order to offer infrastructure services at a global scale. In other words, the virtualization control must cope with interdomain provisioning among independent carriers. As stated in [11], “... virtual networks embedding across multiple infrastructure providers is still a virtually untouched problem.”

Since the dawn of modern telecommunications, there are only two cases of success concerning interdomain CP (*Control Plane*) standardization: SS7 (*Signaling System No. 7*) [14] and BGP (*Border Gateway Protocol*) [15]. No matter GMPLS is being successfully deployed today to offer automatic provisioning of circuits in intradomain scenarios, it completely fails to provide interdomain provisioning so far [16]. One key aspect that has turned SS7 and BGP into successful technologies is the ability of these solutions to satisfy commercial agreements, without compromising internal network manageability or disclosing critical information regarding the network operational status. Indeed, the BGP policy-driven routing mechanism restricts the full exploitation of the underlying topology, as physical connectivity between ASes does not imply reachability [17]. The GMPLS standardization process main concern regards the technical aspects of provisioning control. Commercial relations between carriers and the economic aspects of those relations were completely ignored by GMPLS.

GMPLS-controlled WRPNs are already being deployed commercially. However, the available circuit provisioning techniques still can not fully exploit the potential of WRPNs. L1 optical circuits, better known as LPs (*Lightpaths*), are calculated by RWA (*Routing and Wavelength Assignment*) algorithms. In a GMPLS-controlled WRPN, one or more RWA algorithms are implemented in PCEs (*Path Computation Elements*) [18]. RWA algorithms are very complex to design, and the quality of results are proportional to the network DP (*Data Plane*) status information (made available by the CP) and by the time available to calculation. Over the last fifteen years, the RWA problem has been extensively researched. Despite all advances in the field, the state of the art RWA algorithms still do not deliver the desired performance levels for WRPNs on support of future Internet services. In this case, LPs tend to have shorter duration, may be sensible to setup delay and, most important, may required assured QoS even in case of network failures. Current RWA algorithms can not offer a reasonable trade-off balance between speed and blocking probability, considering the available data from GMPLS CPs and all-optical monitoring devices.

Clearly, the future of the Internet relies in the successful development and deployment of NGNs. And NGNs, to succeed in their mission, must be supported by intelligent, global network infrastructures that will bring to the game new functional capabilities. In order to make these intelligent infrastructures a reality,

1.1. Motivations

5

two challenges must first be overcome: how to organize multiple L1 carriers to render them capable to provide, at an intradomain level, the same service levels that can be achieved internally to each transport network; and also, how RWA algorithms can be enhanced to fully exploit all the potential of WRPNs. This thesis describes three works aimed to address these questions.

The first proposal presents a preliminary architecture to manage and control the services and resources of a set of L1 transport networks, allowing many NGNs to contemporaneously benefit from the multidomain physical infrastructure, seamless and effortlessly. By using network virtualization techniques, NGNs are able to interact with the physical underlayer as it was a dedicated, single domain infrastructure. All L1 infrastructure group members have a strong business relationship between them, and exchange a limited amount of information (including monetary cost of resource allocation) that is enough to provide advanced services such as QoS guaranteed point-to-multipoint circuits.

The second contribution regards the development of a framework that enables the creation of pseudo virtual circuits with assured QoS, using exclusively open source software and commodity PC hardware. The framework is not a simple collection of network tools that are bundled together, but a highly integrated set of systems that performs well-defined tasks. An embedded CP permits the automatic provisioning of circuits, yet offering a great amount of flexibility. The framework is available as live distributions, allowing the configuration of a physical router or even an entire virtual network in few minutes. It is designed to serve as a QoS platform to allow rapid NGN prototyping, and also as an advanced learning tool for graduate networking courses.

The third proposal consists in the development of advanced, survivable RWA algorithms for WRPNs. The main goal of the proposed algorithms is to provide absolute levels of QoS, and guarantee that no service disruption will occur even in the event of single network failures. The proposed RWA algorithms use a highly parallelizable path computation engine and simple heuristics in order to be fast and maximize the resource utilization, therefore minimizing the blocking probability of future LP setup requests.

1.1 Motivations

- The current inability of the GMPLS architecture to fulfill the requirements of interdomain LP provisioning, due to, among other reasons, the intrinsic lack of economic aspects support and commercial agreements considerations.
- The nonexistence of virtualization architectures that allow a substrate to be composed of multiple domains.
- The unavailability of a framework exclusively based on open source

software and commodity PCs to provide pseudo virtual circuits and QoS.

- The incapacity of state of the art RWA algorithms to satisfy the demanding requirements of future Internet services, when concerning LP provisioning.

1.2 Contributions

- A preliminary architecture to enable advanced, future Internet services through L1 multidomain virtualization techniques.
- The development of an open DS-MPLS framework and a flexible software router to aid NGN prototyping, that could also be used as a tool for advanced networking learning.
- The creation of three new RWA algorithms, that introduce novel techniques to perform better than state of the art RWA algorithms, considering processing time and blocking probability as performance metrics.

1.3 Text Organization

Chapter 2 discusses the technology background involving the works introduced in this thesis. Chapter 3 introduces the TNVE architecture; related works in the area along with current and future business models are also presented. Chapter 4 details the open DS-MPLS framework and the developed DS-MPLS router. Chapter 5 introduces the novel RWA algorithms specifically envisioned to WRPNS in support of future Internet services. The state of the art RWA algorithms are initially depicted, followed by detailed descriptions of the proposed algorithms and the evaluation process. Finally, in Chapter 6 conclusions are drawn.

Chapter 2

Background

This chapter presents an overview of the main concepts and technologies involved in this work, in order to provide the appropriate basis to a deep comprehension of the proposals presented in this thesis. The first section reviews the optical transmission systems and introduces the cutting-edge, all-optical networks. The second section discusses two of the most important attempts to introduce QoS in the Internet, namely IntServ and Diffserv. Next, the third section introduces the MPLS switching, while section four discusses integration models for DiffServ and MPLS. The fifth section overviews the GMPLS architecture and its main protocols, and finally the sixth section briefly introduces some network virtualization concepts.

2.1 Photonic Networks

In the early '70s, the first optical transmission systems started to be deployed commercially. At that time, optical networks were formed by point-to-point links which transported only one signal (for example SONET/SDH) per optical fiber, modulated in only one wavelength (usually referred by the Greek letter λ). To cope with the rapidly increasing demand for bandwidth, a whole set of strategies to boost the bulk transmission capacity of optical telecommunication systems were proposed. The solutions ranged from simple deployment of new fibers to more intricate ones, such as increasing the bit rate of TDM (*Time Division Multiplexing*) channels. These were not practical solutions, considering the costs associated with fiber deployment and the technical difficulties regarding increasing the transmission capacity of optical carriers. This scenario has motivated the development of the WDM (*Wavelength Division Multiplexing*) technology. In a WDM optical network multiple channels modulated in different wavelengths are multiplexed and then contemporaneously transmitted in a single fiber. The WDM principle is analogue to FDM (*Frequency Division Multiplexing*), that allows the

contemporary transmission of a number of signals modulated in different carriers without overlapping. Therefore, the original transmission capacity of every single fiber is multiplied by the number of channels supported by the WDM system [19].

The main components of WDM systems are discussed in the following subsections.

2.1.1 Optical Fibers

Fiber optics are used to guide wavelengths between optical network devices with a minimum level of attenuation (signal loss), which varies with fiber quality and length. Fibers are formed by (at least) two layers of different type of silica glass, named core and cladding. Doping techniques are used during the manufacturing phase of modern fibers, in order to considerably alter the index of refraction of silica glass. Due to the lower index of refraction of the cladding, the optical signal is transmitted in the interior of the core at $\frac{2}{3}$ of its velocity in vacuum, by the principle of total internal reflection.

Fibers can be divided in two categories: multimode and monomode. WDM systems use monomode fibers, due to the huge bulk transport capacity and intrinsic low loss of these fibers. Monomode fibers are carefully designed and crafted to the specific use of certain regions of the optical spectrum that present the lowest levels of attenuation. These regions are called *windows*. Nowadays, the windows more commonly used by WDM systems are the S-band (1460 to 1530 nm) and the C-band (1530 to 1565 nm) [20]. The L-band (1565 to 1625 nm) is of particular interest to cutting-edge DWDM (*Dense Wavelength Division Multiplexing*) systems.

2.1.2 Optical Transmitters and Receivers

Light emitters and detectors are active devices that are found in opposing extremities of optical transmissions systems. Light emitters are transmitters that convert electric signals in light pulses. On the other hand, light detectors do exactly the opposite, i.e., transform received light pulses in electric signals.

Two classes of light emitting devices are used in optical transmission systems: LEDs (*Light Emitting Diode*) and semiconductor lasers. WDM systems are designed with semiconductor lasers, due to the precise wavelength tuning, narrow band spectrum, high optical power and *chirp* control (signal frequency variation over time). Depending of the type of the semiconductor laser used, wavelengths of optical signals must first be adjusted before being injected in fiber links. This task is performed by an optical device called *transponder*.

At the reception end, it is necessary to recover the information carried in the electrical domain before being hauled by WDM transmission lines.

2.1. Photonic Networks

9

Photodetectors are employed for this task. Two types of photodetectors are vastly used: PIN (*Positive-Intrinsic-Negative*) photodiodes and APD (*Avalanche PhotoDiode*). While PINs are cheap and robust, APDs have greater precision and reception sensibility.

2.1.3 Physical Impairments and Regenerators

Optical signals traversing the network through optical fibers and devices are subject to QoT (*Quality of Transmission*) degradation by a number of physical layer (or transmission) impairments. Among those, the most relevant factors are chromatic dispersion, PMD (*Polarization Mode Dispersion*), ASE (*Amplified Spontaneous Emission*) and the so called nonlinear effects.

Chromatic dispersion [21] is not an issue in modern WDM systems anymore. It can be compensated on a per-link basis, during the WDM network design phase. PMD compensation [22] in turn is very difficult to perform, and is the principal challenge in the deployment of cutting edge 40 Gb/s and beyond systems. In today’s production networks, PMD effects over QoT are usually avoided by setting up an upper bound for the length of LPs, although it is far from optimal. However, the maximum allowed LP length constraint tends to be relaxed, due to quality enhancements in optical fibers (smaller PMD parameters) and late advances in all-optical PMD compensation [23].

ASE degrades the OSNR (*Optical Signal to Noise Ratio*), and is one of the dominant impairments at any bit rate. ASE noise accumulates as the optical signal traverses a path, and saturation effects may disturb the effectiveness of signal amplification, influencing the BER at the receiver. De-multiplexing filters can strengthen the ASE noise leading to linear crosstalking as well. ASE noise can be acquired either by approximate calculation (using information directly from network elements such as optical input power and losses) or by measuring. In the later case, the WDM network must be equipped with OPM (*Optical Performance Monitoring*) devices [24].

Nonlinear impairments [25] strictly depend on optical power and are the most difficult ones to be treated. Examples are stimulated Raman and Brillouin scattering, four-wave mixing, self phase and cross phase modulation. They can substantially affect the performance of very dense DWDM systems due to nonlinear crosstalking.

To compensate the degradation of light pulses, regenerators are traditionally placed along fiber links. Regenerators are devices formed by optical and electronic components, capable of reamplifying, reshaping and retiming optical signals. This operations is called 3R (*Reamplification, Reshaping and Retiming*) regeneration. Each single wavelength is regenerated individually, involving OEO conversion. Usually, optical signals can be transmitted up to 120 km without amplification. For distances greater than 600 km amplified signals become noisy

and distorted, thus they must be reshaped and retimed [26].

2.1.4 Optical Amplifiers

The technology breakthrough that made WDM transmission lines a reality was the optical amplification advent. Optical amplifiers are devices that, diversely from regenerators, amplify all wavelengths at once, without conversion to the electrical domain. By far, the most used type of optical amplifiers is EDFA (*Erbium Doped Fiber Amplifier*). Using a *pump laser*, a light pulse is injected in the erbium doped fiber. This light pulse stimulates the erbium atoms to release photons in wavelengths around 1550 nm.

The main performance parameters of optical amplifiers are the optical gain, the noise level and the saturation power.

2.1.5 WRPNs

Traditionally, optical networks are formed by WDM transmission lines and OLTs (*Optical Line Terminals*), that perform (de)multiplexing and OEO conversion at each hop in order to add, drop or forward data. These types of networks are known as opaque networks. All the complexities of the optical layer are dealt during the design and deployment stages, and they are not taken into account during provisioning. However, such networks have issues with cost and power efficiency, and cannot leverage modern applications due to slow provisioning time (days, even weeks). Moreover, OEO conversion is the bottleneck for routing at the optical layer. Indeed, the electronic switching matrix of opaque network elements do not have enough processing power to electronically route data hauled by modern WDM systems, that can carry dozens of channels in a single fiber at speeds up to 40 Gb/s per channel.

WRPNs are slowly been adopted to circumvent the limitations of opaque networks [27]. WRPNs are composed of PXC (*Photonic Cross-Connects*, also known as All-Optical Cross-Connects) that are able to perform OCS (*Optical Circuit Switching*) entirely in the optical domain, without the need of OEO conversions. A PXC can be basically divided in two parts: an all-optical switching matrix and a port complex, as illustrated in Figure 2.1. The switching matrix is responsible to route incoming wavelengths from the PXC input interfaces to the output interfaces. These interfaces form the port complex, which connects the PXC to other network elements using optical fibers. Border PXC are equipped with OADM (*Optical Add/Drop Multiplexers*) to allow vertical data exchange between the optical cloud and the clients of the optical layer. Data about to be carried by a WRPN are converted to the optical domain at the ingress PXC, and remain in the form of an optical signal while traversing the network.

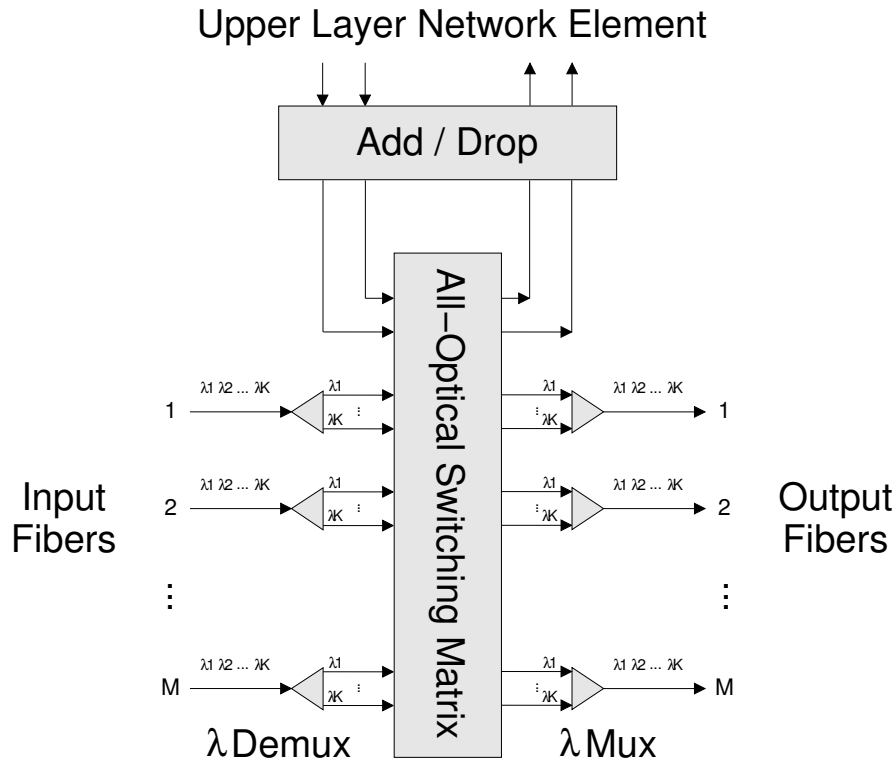


Figure 2.1: A PXC formed by M fibers, each one carrying K wavelengths.

The main advantages and drawbacks of WRPNs come from the same reason: the absence of OEO conversion. Data are carried at high throughput transparently by LPs, thus they can have any format or rate. On the other hand, optical signals are not regenerated in WRPNs. Therefore, optical transmission impairments [28] are accumulated while wavelengths traverse the network. These physical layer impairments degrade the optical signal quality and indirectly affect the BER of traffic being carried. While opaque networks usually have point-to-point or ring topologies, WRPNs tend to be meshed, with longer links and larger number of nodes. In this scenario, a given LP may not be eligible to haul traffic due to poor QoT, even if there are abundant available resources.

The set of LPs established in a WRPN reflects the topology of the client networks of the WRPN. Figure 2.2 shows an IP network on top of a WRPN. In this scenario, three LP were previously setup (A-B, A-C and B-C). The LP topology formed by these circuits is in fact the upper layer IP topology (where all routers are adjacent), that is different from the WRPN one (a line topology). Thus, the LP topology is also known as the *virtual topology*.

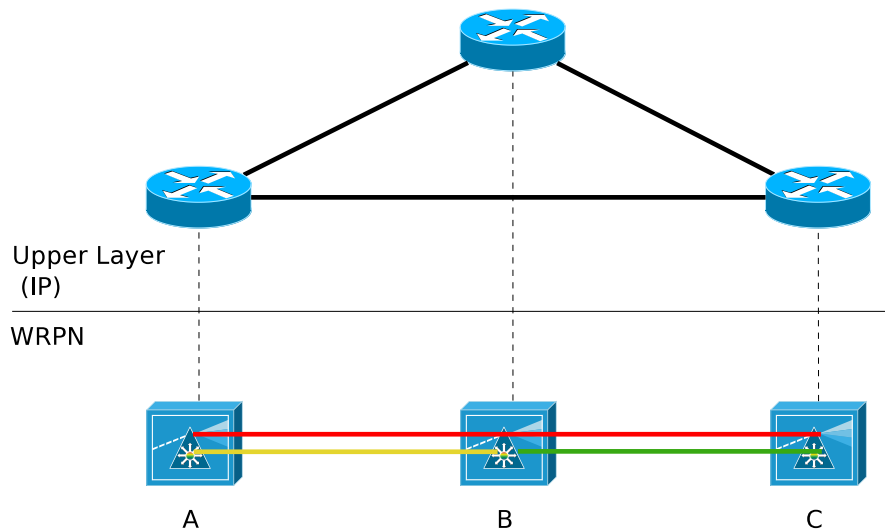


Figure 2.2: An IP network on top of a WRPN.

2.1.6 RWA

In order to establish a LP in a WRPN, it is necessary to select not only a path between the ingress and egress PXC's, but also a wavelength to modulate the optical signal on that path. This problem is known as RWA . RWA algorithms can be divided in two main categories:

Static RWA

RWA algorithms of this class are useful when the traffic matrix is known before the WRPN becomes operational. Hence, the static RWA problem is solved by offline algorithms. Usually, their goal is to minimize the number of wavelengths necessary in order to establish a certain set of LPs, given a WRPN with a known topology. Another alternative is to maximize the number of LPs that can be setup, considering a fixed number of wavelengths.

Dynamic RWA

In this case, the traffic matrix is not known a priori. LP setup and/or teardown requests arrive in an (usually) unpredictable fashion. Thus, dynamic RWA problem is solved by online algorithms. The objective of these algorithms is to satisfy the incoming requests, and if possible to minimize the blocking probability of future requests.

The RWA problem is subject to a number of constraints that do not affect the routing process in packet networks. As a result, it is possible that a LP setup

2.1. Photonic Networks

13

request is blocked even if there are unused network resources. These additional restrictions can be sorted in three groups:

Wavelength-continuity constraint

Data being carried by a LP in a WRPN remain in the optical domain all the time. As the all-optical wavelength conversion (without electronic processing) is still an immature technology, the same wavelength must be assigned to all links that together form the LP path.

Diversity constraint

This restriction is taken into account only by survivable RWA, i.e., when the LP to be setup is intended to carry protected data. Two or more LPs are called “diverse” if they are not subjected to be disrupted simultaneously by a network failure. In most of the cases, LP diversity means link-disjoint paths.

Physical layer impairments

As optical signals are not regenerated anymore, the physical impairments discussed in subsection 3.4 can not be ignored when establishing LPs in WRPNs. Algorithms that consider physical layer impairments in order to provide LPs with QoT are known as IA-RWA (*Impairment Aware Routing and Wavelength Assignment*).

To achieve optimal solution, the RWA problem is formulated using MILP (*Mixed-Integer Linear Programming*), which requires a high computational power even for not so complex topologies. MILP techniques are used to solve offline the static RWA problem, considering that time is not a hard constraint. Online RWA algorithms, on the other hand, must have a total run time as short as possible, as the time necessary to satisfy a LP setup request in a WRPN can be as low as some tens of seconds. To accomplish this task, traditionally the RWA is divided in two sub-problems. The routing and the wavelength assignments are solved independently, using heuristics that provide sub-optimal results in feasible times. The most important wavelength assignment heuristics today are FF (*First-Fit*), LF (*Last-Fit*), BF (*Best-Fit*), MU (*Most-Used*) and RP (*Random-Pick*). The routing subproblem can be solved using SPF (*Shortest Path First*) algorithms, such as the Dijkstra algorithm. Some approaches calculate routes offline for every possible source-destination pair, leaving only the wavelength assignment to be carried out after a request for LP setup has arrived. When only one path for every source-destination pair is calculated, the heuristic is called FR (*Fixed Routing*). When more than one path (three is a common value) is precomputed for all source-destination pairs, this approach is known as FAR (*Fixed-Alternated Routing*) [29, 30].

2.2 Internet and QoS

The Internet was conceived to deliver packets according to the BE (*Best Effort*) forwarding paradigm. BE Internet service delivery can be defined as FIFO (*First-In First-Out*) queuing and LIFO (*Last-In First-Out*) tail-first dropping. The QoS (in terms of packet delivery delays and drops due to buffer overflow) of BE service depends not only on the network actions (which the network can control), but also on the offered load (which the network cannot control). Thus, in BE service, the network tries to forward all packets as soon as possible, but cannot make any quantitative assurances about the QoS delivered [31].

The first attempt to offer some sort of relative QoS provisioning is described in the original "DARPA Internet Program Protocol Specification" [32], later updated in [33]. These documents respectively introduce and redefine the TOS (*Type Of Service*) octet of the IP header, which guide the selection of the actual service parameters when transmitting a datagram through a particular network. The TOS octet is shown in Figure 2.3.

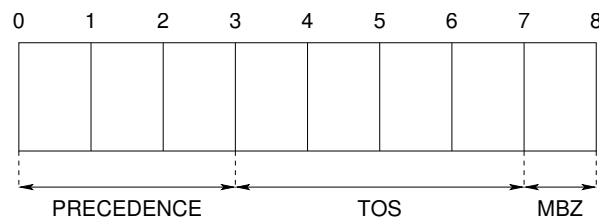


Figure 2.3: The TOS octet, part of the IP header.

The TOS octet is composed by three fields, namely

Precedence

Denotes the "importance" of the datagram, and specifies seven priority levels:

- Routine
- Priority
- Immediate
- Flash
- Flash Override
- CRITIC/ECP
- Internetwork Control
- Network Control

2.2. Internet and QoS

15

TOS

Denotes how routers should treat the IP packet, making tradeoffs between throughput, delay, reliability, and cost. Five values (out of the sixteen allowed by 4 bits) have been defined:

- Normal Service
- Minimize Monetary cost
- Maximize Reliability
- Maximize Throughput
- Minimize Delay

MBZ (*Must Be Zero*)

Unused field, must always be set to zero.

No matter that [33] defines in detail the scope and properties of the TOS octet, it does not specify how routers should enforce traffic prioritization. Indeed, the TOS facility never has been really adopted in large scale. Even when the TOS field of IP packets was filled by hosts (only Unix systems by that time), the vast majority of routers completely ignore it during the routing process.

As new QoS-sensitive Internet applications (like multimedia content delivery) started to appear, the need for a *QoS-aware* Internet arose. During the last two decades the industry and academic communities have proposed a number of solutions, with different levels of success, to render the Internet a better platform to leverage QoS-sensitive applications. Two of these proposals are briefly described in the next subsections.

2.2.1 IntServ

The Integrated Services model, or simply IntServ [34], aimed to provide end-to-end QoS between host applications using reservation techniques. It extends the original Internet architecture introducing new components and functionalities, without compromising the standard BE forwarding of IP networks. IntServ defines methods for identifying traffic flows, and controls the end-to-end packet delay on a per-flow basis. This is achieved by a mechanism called *controlled link-sharing*. The IntServ model introduces the following services:

Guaranteed Service

Used by applications that are sensitive to delay and jitter, thus requiring real-time service delivery. Each router in the path must reserve network resources in order to guarantee absolute QoS levels for the requested service.

Controlled-Load Service

Used by applications that are not strictly delay sensitive, but can not perform well in overload conditions. Controlled-load offer low queueing delay and low packets loss services to application, without implementing complex per-flow control mechanisms.

The IntServ architecture introduces the building blocks later used by many QoS frameworks for IP networks. The four main components of the IntServ architecture are:

Scheduler

manages the packet forwarding of different flows using a set of queues and timers. Schedulers perform per-flow statistics of forwarded packets (either measuring or estimating the outbound traffic), also usefull for the admission control.

Admission Control

determines if the QoS level specified in a new service request can be granted without compromising the other preestablished flows. Also, it is in charge of administration tasks such as the service request authentication and billing.

Classifier

using IP header information, its task is to arrange packet flows in groups. Different flows belonging to a same group, called *class*, receive the same forwarding treatment. A class can be locally abstracted. For example, backbone routers can aggregate many flows in few classes, while access routers can map a single flow per class.

Resource Reservation Protocol

necessary to manage flow requests and perform maintenance of flow status along the path. For this purpose, IntServ uses the RSVP (*Resource ReserVation Protocol*) [35].

2.2.2 DiffServ

By the time of IntServ standardization, concerns about scalability in large scenarios have appeared. Indeed, to keep record and provide maintenance of flow-states is a heavy burden to routers, which must have at disposal a great amount of storage capacity and processing power. The Differentiated Services architecture (shortly DiffServ [36, 37]) was designed with scalability as the main concern, and is intended to be simpler than Intserv. The per-flow service was substituted with per aggregate service, and the complex processing moved from the core to the edge of networks [38].

2.2. Internet and QoS

17

Diversely from IntServ that uses RSVP to allocate resources in real-time, the Diffserv model provides services using QoS parameters described in preestablished SLAs (*Service Level Agreements*) between customers and ISPs. Accordingly to the traffic aggregate that a packet belongs to, it receives a mark that is storage in the TOS octet, now renamed as the DS field [39]. The mark is placed in the first six bits of the DS Field, a subfield called DSCP (*DiffServ Code Point*). Each DSCP value indicates a different packet forwarding treatment at each router, which are called PHB (*Per-Hop Behaviour*). The DS Field structure is shown in Figure 2.4. The two last bits of the DS field were defined as "Currently Unused" by the DiffServ architecture and should be ignored by DS-routers, but at the present time these two last bits are used to incorporate ECN (*Explicit Congestion Notification*) to IP and TCP [40].

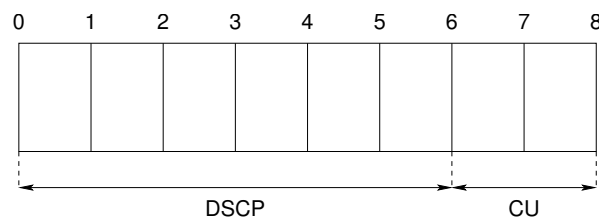


Figure 2.4: The DS Field structure (redefinition of the TOS octet of the IP header).

Initially, three types of PHBs have been defined. Later on, configuration guidelines have been made available [41]:

Default (DE)

The standard BE forwarding paradigm. Packets with the a DSCP value equal to 000000 or packets with an unrecognized value are forwarded using the DE PHB.

Expedited Forward (EF) [42, 43]

This PHB provide a low-loss, low-delay, low jitter service. Its DSCP value is 101110, and can be implemented with a short buffer, priority queue. More complex schedulers such as DRR (*Deficit Round Robin*) and WFQ (*Weighted Fair Queuing*) can also be used to implement EF services. Being the PHB with the highest QoS profile specified by the DiffServ model, it is also the most expensive. Therefore, EF services are often called *premium* services. Non-conformed EF traffic is silently dropped.

Assured Forward (AF) [44, 37]

This PHB does not consider upper bounds for QoS parameters like delay or jitter. Instead, it forward packets with a high probability of delivery, considering that the traffic conforms to the contracted rate. Traffic that do meet SLAs are allowed, but are subjected to be discarded. AF services

can be described as "Better-than-Best-Effort". Four class of services have been defined in the AF PHB, and each one has three possible discard levels known as drop-precedence. The higher the drop-precedence, the higher is the probability that the packet is discarded. The AF PHB is usually implemented with RED (*Random Early Detection* [45] or one of its variants, like RIO (*RED with In/Out*) [46]. Table 2.1 depicts the defined AF classes and associated DSCP values.

Table 2.1: DSCP values for all AF X classes and Y drop precedences.

Drop Prec.	AF 1Y	AF 2Y	AF 3Y	AF 4Y
AF X1 (Low)	001010	010010	011010	100010
AF X2 (Med.)	001100	010100	011100	100100
AF X3 (High)	001110	010110	011110	100110

A DS-enable router performs two additional functions: packet classifying and conditioning. The former is responsible to correlate a packet with an PHB, using the IP header data (the DS Field can be overridden, if previously set by another domain). The later uses the result of packet classification as input to enforce SLA profiles on traffic aggregates. The DiffServ Traffic Conditioner is composed by a meter, a marker, a shaper and a dropper. The functional flowchart of the DiffServ Model is illustrated in Figure 2.5.

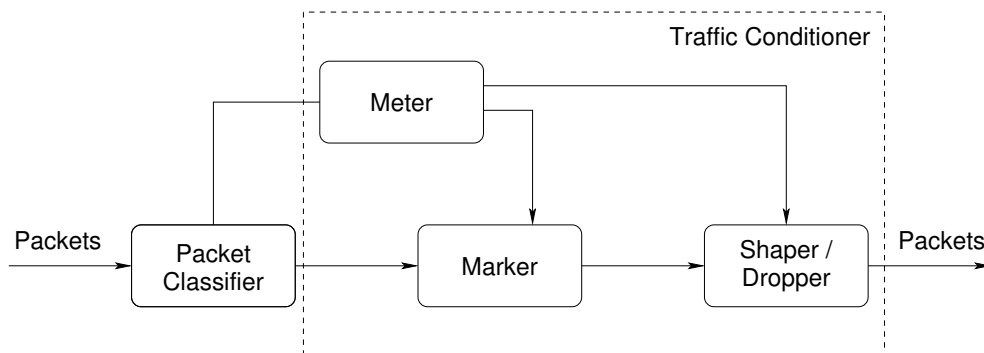


Figure 2.5: Functional flowchart of the DiffServ Model.

2.3. MPLS Switching

19

2.3 MPLS Switching

Since the late '80s, network vendors have began to bundle new functionalities in IP routers on an attempt to cope with the ever growing Internet traffic. Due this increasing complexity, the gap between layer 3 routers and layer 2 switches were also increasing, when comparing their forwarding capacities. By the early '90s, ATM (*Asynchronous Transfer Mode*) networks were used to haul IP packets at higher speeds and longer distances, despite the high overhead of IP-over-ATM encapsulation models like LANE (*LAN Emulation*) and MPOA (*MultiProtocol Over ATM*). This scenario paved the way for the development of technologies that brought to the network layer the connection oriented, packet forwarding paradigm, by means of label switching. In 1997, the IETF (*Internet Engineering Task Force*) has created a working group to standardize these technologies. The new standard was baptized with the vendor-neutral name of MPLS (*MultiProtocol Label Switching*) [47].

In traditional BE delivery, IP packets are forwarded by routers using exclusively the information in the IP header. Almost always, only the destination field is used. In a MPLS network, the IP header data is used only at ingress routers to map packets in FECs (*Forwarding Equivalence Classes*). A FEC describes a set of packets that have similar or even identical characteristics. Packets belonging to the same FEC are routed using the same criteria. Each packet entering the MPLS cloud is associated with a FEC and receives a *label*, that will be used by the routing process to forward the packet throughout the network. Figure 2.6 depicts the possible combinations of link layer and network protocols that can be integrated by the MPLS standard.

IPv6	IPv4	IPX	AppleTalk	Network Layer Protocols
Label Switching				
Ethernet	FDDI	ATM	Frame Relay	Data Link Layer Protocols
			PPP	

Figure 2.6: Link layer and network protocols that can be integrated by the MPLS standard.

Some link layer technologies (like ATM and *Frame Relay*) have reserved fields

in their cells or frames to carry the label information. When this is not possible (for example in *Ethernet* networks), a *shim header* is introduced between the headers of layers 2 and 3. The shim header has 32 bits, and is composed by four fields:

Label

Label value (20 bits).

EXP

Experimental Use (3 bits), now renamed to TC [48].¹

S

Bottom of Stack (1 bit).

TTL

Time to Live (8 bits).

Figure 2.7 shows the shim header between the link layer and network headers.

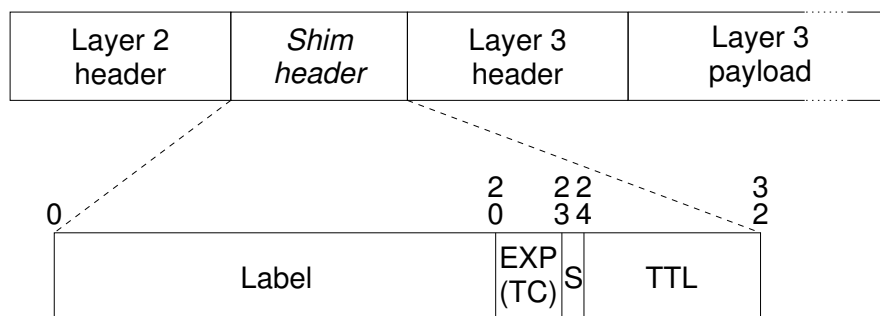


Figure 2.7: Shim header between the link layer and network headers.

Inside the MPLS cloud, each LSR (*Label Switching Router*) has a database called LFIB (*Label Forwarding Information Base*), which contains all the information necessary to forward packets. When a packet arrives at a core LSR, its label is used to perform a lookup at the LFIB. Each incoming label is associated to an outgoing label and interface, that are used to quickly forward the packet. When a packet leaves the MPLS core, the shim header (if present) is removed. The path used by the packets of a given FEC to cross the MPLS network is called LSP (*Label Switched Path*).

¹As the TC designation is still new and not widely used, and also by the fact that the term *TC index* is used in Linux traffic control (which in the context of this thesis is closely related to the EXP field), it has been decided to still use the previous ‘EXP’ terminology to refer to the second, 3 bits field of the shim header.

2.4 DS and MPLS integration

By using DS alone, ISPs are able to prioritize premium traffic applying distinct PHB to different traffic aggregates. However, the DS model essentially offers relative QoS, and the only way to meet strict SLA requirements is to provide enough resources to avoid chronic congestions. MPLS itself allows ISPs to perform TE (*Traffic Engineering*), balancing the network load and assuring protection and absolute QoS for LSPs. Nevertheless, the concept of CoS (*Class of Service*) differentiation is not defined by the MPLS. If multiple flows are associated with a given LSP (by using one or more FECs), the non-conforming aggregate traffic will be shaped (and eventually dropped) without distinction between the flows carried by the LSP. Therefore, DS and MPLS are complementary technologies, that together allows per-CoS resource allocation and protection, without the need of over provisioning [49].

Two standards were proposed by the IETF to unify DS and MPLS. The first proposal is best known as “DiffServ over MPLS” [50], and specifies two different types of LSPs specially designed to carry DS traffic:

E-LSP (*EXP-Inferred-PHB Scheduling Class LSP*)

The EXP field of the MPLS shim header is used by LSRs to determine the PHB to be applied to the IP packet. Thus, an E-LSP can support up to 8 distinct PHBs, drop precedences included. The mapping between PHB and EXP values can be statically defined or negotiated online, during the LSP signaling process.

L-LSP (*Label-Only-Inferred-PHB Scheduling Class LSP*)

Only one PHB is applied to all packets belonging to the FEC. The PHB is associated using exclusively the label during the LSP setup phase. In case the PHB is AF, the drop precedence information is stored in the EXP field of the shim header.

The second proposed unification model is called “DiffServ-aware MPLS TE”, or simply “DS-TE” [51, 52]. Some definitions used by DS-TE are:

Traffic Trunk

an aggregation of traffic flows of the same class which are placed inside a LSP.

CT (*Class-Type*)

the set of Traffic Trunks crossing a link, that is governed by a specific set of bandwidth constraints. CT is used for the purposes of link bandwidth allocation, constraint based routing and admission control. A given Traffic Trunk belongs to the same CT on all links.

TE-Class

a tuple formed by a CT and a preemption priority allowed for that CT.

DS-TE defines eight TE-Classes, (from TE0 to TE7), eight CTs (from CT0 to CT7) and eight priority levels (from 0 to 7). Each ISP has the flexibility to specify how each TE-Class is formed, based on the 64 possible combinations between the available CTs and priority levels. For example, if an ISP wants to be able to preempt BE traffic in favor of voice traffic, but also preempt voice traffic from customer A in favor of voice traffic from customer Z, the follow TE-Classes can be defined:

TE0 [CT0, priority 0]: voice traffic (EF PHB mapped to CT0), customer Z

TE1 [CT0, priority 1]: voice traffic (EF PHB mapped to CT0), customer A

TE2 [CT1, priority 7]: BE traffic (BE default PHB mapped to CT1)

Two different bandwidth constraint models are proposed to be used with DS-TE: MAM (*Maximum Allocation Model*) and RDM (*Russian Dolls Model*) [53]. MAM assigns portions of the link bandwidth to each defined CT, which are completely isolated from each other. This means that the preemption priorities are considered only between LSPs carrying traffic of the same CT. RDM, in the other hand, allows bandwidth sharing between different CTs. With this model, CT0 traffic (usually mapped as EF) can use the bandwidth allocated to all other CTs (CT7 is mapped as the BE default PHB). While RDM provides efficient bandwidth sharing between all traffic trunks traversing the network, preemption is needed to guarantee bandwidth to all CTs.

2.5 GMPLS Architecture

Lower layer networks usually rely on heterogeneous, non packet-based switching mechanisms, i.e., cannot forwarding data using information carried in packet or cell headers. The ability to perform fast, automatic circuit provisioning in these networks, that currently are the foundation of the Internet infrastructure, are key to reduce CAPEX (*Capital Expenditure*) and OPEX (*Operational Expenditure*) and also to generate revenue. GMPLS extends the original MPLS architecture to support LSP provisioning in networks where switching decisions are based on time slots, wavelengths, or physical ports. Moreover, GMPLS introduces new features, like complete separation of CP and DP, bidirectional LSPs, link bundling, unnumbered links, and forwarding adjacencies [12]. GMPLS can also be seen as an IP-based instantiation of the protocol independent ASON (*Automatic Switched Optical Network*) [54] architecture, proposed by ITU-T (*International Telecommunication Union - Telecommunication Standardization Sector*).

2.5. GMPLS Architecture

23

The GMPLS suite of protocols consists of both new and extended MPLS protocols. The main functions of the GMPLS CP are:

Link Management

Encompass the functionalities of neighbour discovering, link properties correlation and control channel maintenance.

Routing

Relates to resource gathering and dissemination inside the control domain.

Signaling

Cope with LSP provisioning, protection and restoration.

Path Computation

Responsible to find constrained-based paths to satisfy complex LSP setup requests.

The next subsections briefly describe the GMPLS protocols.

2.5.1 LMP

In GMPLS controlled networks, adjacent network elements can be connected by several links. Each link can be a fiber with tens to hundreds of wavelengths channels. To manually identify all channels and to configure their properties at both ends is not a viable option. These tasks, that must be repeated after topology changes, are time consuming and highly subject to error. Moreover, the control and data planes can have different topologies, and even not share the same physical medium. In order to cope with these issues, the LMP (*Link Management Protocol*) [55, 56] has been introduced. It is responsible for control channel maintenance, neighbour discovering and link-properties correlation. The LMP is a point-to-point protocol that uses UDP datagrams to exchange messages with adjacent network elements. To establish an adjacent relationship, at least one control channel must be active between a pair of GMPLS-enabled nodes. The most important tasks performed by LMP are [57]:

- During startup process, adjacent neighbours activate control channels and start to exchange LMP messages to gather information about their identities and capabilities. Once control channels are activated, *Hello* messages are used to perform their maintenance.
- The *Link Discovery* process allows GMPLS nodes to determine the existence, the nature and the connectivity status of their links. Before the Link Discovery process takes place, the only information available are the links local identifiers. Their operational status and remote identifiers are

unknown. This information is gathered by exchanging messages between nodes.

- The *Link Verification* process can be activated at any moment, in order to verify the operational status of links between LMP peers. This process is identical to the Link Discovery process.
- *Fault Isolation* is one of the most important features of LMP, particularly to WRPNs. In case of a fiber cut, all downstream PXC's located after the fiber cut will detect the LoL (*Loss of Light*). Considering that hundreds of LPs can cross a single fiber, a single failure can trigger an *alarm storm*. Thus, traditional methods for layer 2 and 3 link health monitoring are not appropriated. The LMP message-based Fault Isolation process is able to detect the exact point of failure and avoid alarm storms.

2.5.2 OSPF-TE

The OSPF (*Open Shortest Path First*) [58] is a link-state routing protocol, first specified in 1991 to substitute RIP (*Routing Information Protocol*) [59] which was facing scalability problems on the Internet. OSPF introduces functionalities like equal-cost multipath, hierarchical routing, separation between internal and external routes, and enhanced security. It uses LSAs (*Link State Advertisements*) to inform the state of links. LSAs are disseminated through the network using a mechanism called *reliable flooding*. This mechanism guarantees that all routers in an OSPF area will end up with the same set of LSAs, which is called LSDB (*Link State DataBase*).

The OSPF protocol was extended by the MPLS and GMPLS standards to allow the transportation of TE metrics related to properties of links. This extension is known as OSPF-TE [60, 61]. To properly disseminate TE metrics using the OSPF flooding mechanism, a special type of LSA has been created, which is called *Opaque LSA* [62]. TE information carried by opaque LSAs, in this case TE-LSAs, are organized in TLV (*Type-Length-Value*) triples. TLVs are extensible data structures, that can also be nested. Figure 2.8 shows a TLV tripe and its fields.

The OSPF-TE documentation specifies two top-TLVs, *Router Address TLV* and *Link TLV*. While the former carries an IP to uniquely identify the GMPLS node inside a domain, the latter is used as an envelop to carry several sub-TLVs. These sub-TLVs describe TE links metrics like nominal, reserved and unreserved bandwidth, local and remote identifiers, and protection-type parameters. So far, the proposed TE link sub-TLVs are not enough to describe all the information necessary in order to allow the use of advanced IA-RWA algorithms in GMPLS-controlled WRPNs [63].

2.5. GMPLS Architecture

25

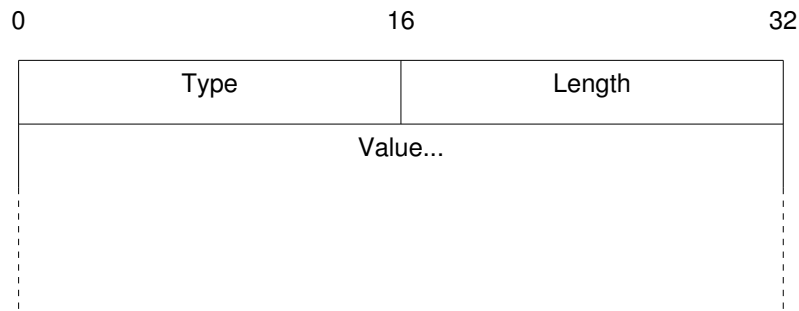


Figure 2.8: A TLV triple.

2.5.3 RSVP-TE

As discussed in subsection 2.2.1, the RSVP protocol was introduced by the IntServ architecture to manage traffic flows, reserving resources along a preestablished path. The GMPLS standard proposed a series of enhancements to the RVSP protocol, to render it capable of performing label distribution and TE link resource reservation. This version of the protocol has been named RVSP-TE [64, 65]. RSVP-TE messages can carry RSVP objects, subobjects and (more recently) TLVs to describe TE link and LSP attributes. The main features of RSVP-TE are:

- The use of PATH and RESV messages to suggest, require and assign labels during the LSP setup phase;
- To allow the use of a explicit route during LSP establishment or rerouting;
- To specify any TE link parameter like bandwidth or protection during the LSP setup phase;
- The use of *Hello* messages to establish a neighbouring relationship between RSVP-TE peers;
- The use of *Notify* messages to propagate information regarding the operational status of LSPs.

More recently, extensions were proposed to the RSVP-TE protocol to support Graceful Restart [66], contiguous, nested and stitched interdomain LSPs [67], P2MP (*Point-to-MultiPoint*) LSPs [68] and MP2P (*MultiPoint-to-Point*) [69] LSPs.

2.5.4 PCE and PCEP

Traditionally, TE-LSPs are established in a MPLS network at the head end router, either by using a CSPF (*Constrained Shortest Path First*) engine to calculate the path or by using an explicit path, specified in a RVSP ERO (*Explicit Route Object*) structure. The introduction of GMPLS allowed the control of non packet switching capable networks, with complex to manage DPs. In such networks, LP provisioning is subject to several issues, mainly:

- CPU-intensive path computation
 - multi-criteria path computation, e.g. physical impairments in WRPNs;
 - link-disjoint backup path computation;
 - minimal cost P2MP trees;
 - global LSP rearrangement to promote network resources optimization.
- unavailability of TEDs (*Traffic Engineering Database*) at border LSRs. TED construction and maintenance is a heavy burden to routers;
- multidomain and multilayer path computation, without full visibility of network topologies;
- the ability to discriminate LSP setup requests generated by different clients.

This scenario led to the introduction of a new entity inside the GMPLS control plane, called PCE (*Path Computation Element*) [18]. One or more PCEs can be found inside a domain, either integrated on LSRs or as dedicated servers. PCEs are responsible to collect LSP setup requests from PCCs (*Path Computation Clients*) and elaborate paths for these requests, which can be performed in a standalone fashion or with collaboration with other PCEs. Once a reply message from a PCE describing the calculated path is acknowledged by a PCC, the LSP can be finally instantiated using standard RSVP-TE signaling. PCC requests are validated in PCEs using local policy components.

The communication between PCC and PCEs, and also between pairs of PCEs, are defined by the PCEP (*Path Computation Element communication Protocol*) [70]. The PCEP is a TCP-based protocol, and defines the following messages that are exchanged between two peers (any combination of PCCs and PCEs):

Open and Keepalive

these messages are used to to initiate and maintain a PCEP session, respectively.

2.6. Network Virtualization

27

PCReq

the PCEP message sent by a PCC to a PCE to request a path computation.

PCRep

the PCEP message sent by a PCE to a PCC in reply to a path computation request. A PCRep message can contain either a set of computed paths if the request can be satisfied, or a negative reply if not. The negative reply may indicate the reason why no path could be found.

PCNtf

the PCEP notification message either sent by a PCC to a PCE or sent by a PCE to a PCC to notify of a specific event.

PCErr

the PCEP message sent upon the occurrence of a protocol error condition.

Close

a message used to close a PCEP session.

PCEP message consists of a common header followed by a variable-length body made of a set of RSVP-like objects that can be either mandatory or optional. PCEP message objects may also include one or more TLVs to describe their contents.

It is important to state that the GMPLS standard defines only the PCE architecture and how the CP entities involved in the path computation should exchange information between themselves. It is out of the scope to specify mechanisms for patch computation, e.g. IA-RWA algorithms and interdomain, TE-LSP establishments.

2.6 Network Virtualization

As stated in [71], “[the concept of virtualization] consists in adding an abstraction layer between users and physical resources, while giving the users the illusion of direct interaction with those resources ... network virtualization is an emerging concept that extends the concept of virtualization from individual nodes (or resources) to entire networks. The main idea consists in the creation of several co-existing logical network instances (or virtual networks) over a shared physical network infrastructure.”. Historically, virtual networks can be categorized in four classes [11]:

VLAN (*Virtual Local Area Network*)

a group of logically networked hosts with a single broadcast domain regardless of their physical connectivity.

VPN (*Virtual Private Network*)

a dedicated network connecting multiples sites using private and secured tunnels over shared or public communication networks. In most cases, VPNs connect geographically distributed sites of a single corporate enterprise.

Active and Programmable Networks

these networks enable the creation, deployment and management of novel services on the fly, accordingly to user demands.

Overlay Networks

a logical network on top of one or more existing physical networks. Overlays in the current Internet are typically implemented in the application layer.

Network virtualization is being considered as a plausible solution to overcome current Internet ossification. In this context, the role of ISPs is divided by two distinct entities: infrastructure providers and service providers. The former is responsible to manage the physical resources, supporting virtual networks created by the latter. The fundamental principles of network virtualization are coexistence, recursion, inheritance and revisitation. Moreover, the following goals must be considered when realising network virtualization in support of NGNs [11, 72, 73]:

- flexibility;
- manageability;
- scalability;
- isolation;
- stability and convergence;
- programmability;
- heterogeneity;
- legacy support.

To reach these goals, many aspects of network virtualization must be revised and enhanced. Some of them are completely unexplored, as the establishment of virtual networks across multiple infrastructure providers. The following research challenges are detailed in [11] and references within:

- standard interfaces;

2.6. Network Virtualization

29

- signaling and bootstrapping;
- resource and topology discovery;
- admission control and policing algorithms;
- node and link virtualization definitions;
- naming and addressing;
- mobility management;
- configuration, monitoring and failure handling;
- security and privacy;
- interoperability issues;
- network virtualization economics.

Chapter 3

Transport Network Virtual Environments

This chapter introduces a new paradigm on leveraging future Internet services, through the use of L1 virtualized network concepts. First, the related works are presented. Then, the TNVE (*Transport Virtual Network Environment*) is introduced. The next sections discuss the TNVE business model and application scenario, as well as the architecture of the TNVE main element. The chapter is concluded with a case study of a TNVE functionality.

3.1 Related Works and TNVE presentation

Nowadays, a number of future Internet infrastructure support projects is being carried out, financed by top organizations around the world. The main motivation for these ambitious research initiatives is the realization that no revolutionary concept for the future Internet can be validated without being tested in large-scale environments. Indeed, the earlier a novel idea is confronted with the complexities of a large sized scenario, the better. Moreover, the use of network resource virtualization techniques to design and implement considerably large, experimental facilities was set as a requirement by the leading actors on infrastructure support projects. Among them, the NSF (*National Science Foundation*) program GENI [74] in the USA, the NWGN (*New Generation Network*) and AKARI [75] programs in Japan, and the FEDERICA [76] initiatives in Europe can be cited, to name a few.

PlanetLab [77] was the first globally distributed research community testbed that used the network virtualization paradigm. It started back in 2003, and currently consists of 1111 nodes at 515 sites. Each physical node on PlanetLab hosts a number of virtual nodes, and a collection of virtual nodes form a *slice*. Nodes within a slice form an overlay network on top of the Internet. Virtual nodes

are freely programmable, and slices are allocated and managed independently by all participating members. Local site administrators are responsible for add or remove virtual nodes in a slice. Being a network with the Internet itself as the underlay, PlanetLab is not suitable for lower-layer protocols and architectures research. Other limitations include the use of PC commodity hardware and the impossibility to incorporate other link and node technologies. The GENI program extends the PlanetLab vision coping with the previously mentioned limitations. New features include: a richer set of node technologies, support for layer 1 and 2 slicing (by exposing low-level link behavior), instrumentation of all network components for measurement purposes, and federation of resources. GENI allows the creation of clean-slate experiments with their own independent management protocols.

The FEDERICA project proposes an architecture similar to the GENI initiative, that also applies the virtualization principles of substrate, slicing and federation. FEDERICA was a 30 months European Commission funded project, that started in January 2008. Its goal is to provide an infrastructure for future Internet research activities accomplished by hosting virtualized facilities in the substrate nodes. One of FEDERICA main design principles is experiment reproducibility, therefore it is key to provide a controlled environment that is flexible and reliable, with simple and easy to use management solutions. Indeed, finding the proper management concept for virtualization-capable networks and theirs services is currently a real challenge. The main feature of FEDERICA is to allow users to fully configure and manage the resources of their own slices in a completely isolated fashion, without affecting the physical infrastructure operation and, thus, impacting on slices of other users. For this purpose, a SOA-based (*Service Oriented Architecture*) model has been designed. The FEDERICA physical substrate is built on top of the GÉANT2 [78] Pan-European backbone network, that interconnects 34 countries through 30 NRENs (*National Research and Education Networks*), using multiple 10Gbps wavelengths. The topology is composed of 13 physical sites. FEDERICA nodes facilities include programmable high-end layer 2 and 3 network elements, multiprotocol switches and PC-based virtualization-capable devices.

At the physical layer, the Internet will continue to grow as a set of interconnected autonomous domains, due to technical (like scaling requirements), political and economic reasons. Hence, a virtualized worldwide data hauling infrastructure must cope with L1 interdomain relationships, mobilizing resources from multiple, heterogeneous carrier networks [79]. The challenge of providing a virtual network environment that spans multiple transport networks is an almost untouched problem [11]. This motivated the design of a preliminary TNVE architecture to tackle with this very relevant issue, which must be overcome in order to leverage NGN advanced features (such as end-to-end QoS) at a global scale. The main goal of our architecture is to allow client NGNs to build their own managed, easily customizable, on demand setup, underlay transport network,

3.2. Internet and TNVE Business models

33

providing what we defined as NlaaS (*Network Infrastructure as a Service*). The concepts introduced by GENI and FEDERICA were taken into account during the elaboration of the TNVE architecture to ensure the virtualization benefits. Additionally, novel concepts were envisaged to allow the architecture to hide the complex mechanisms of L1 resource allocation and the multidomain intricacies, and to export interfaces that allow client NGNs to request services in a seamless manner. By "services" we mean not only traditional high bandwidth point-to-point circuits, but also P2MP and MP2P ones, with survivability and QoS guarantees. Moreover, the proposed TNVE architecture breaks the paradigm that transport networks must not touch the payload, and limit themselves to just haul bits. This way, NGNs can focus on the end users needs, relying on the underlay TNVE infrastructure to realize on-the-fly bulky operations at interface speed, like cryptography and compression. All of this taking into account the economic aspects of service provisioning, which has a crucial impact on multidomain TE. By requesting infrastructure services from the TNVE instead of directly managing interactions with L1 carriers, NGNs have a number of benefits. Without the need of concerning about the underlay infrastructure anymore (implementing multiple data and control protocols, monitoring techniques, on-the-fly data manipulation, buying a series of telecommunication devices), NGN ISPs can save on CAPEX and OPEX, and little effort is needed to add capacity as necessary. Using the TNVE, infrastructure services are provided quicker with less resource utilization, which also leads to CAPEX and OPEX savings and easier advanced services provisioning.

3.2 Internet and TNVE Business models

The current Internet business model is basically composed of three main actors: customers, network providers and transport networks, as shown in Figure 3.1.

Customers can be divided in end users, which are connected to the Internet through access networks and LANs, and applications providers that use the Internet as a business platform (e.g., online stores and content providers). Customers rely on network providers to obtain Internet connectivity (and therefore, global reachability). ISPs and research networks are examples of network providers, which as a whole form the collection of ASes that is the Internet itself. Transport networks, on its turn, are the telecommunications carriers responsible for long-haul data conveyance. Horizontal and vertical contractual relationships between actors (of equal and different types, respectively) are called SLAs.

Unlike SLAs between two network providers, SLAs between network providers and transport networks usually describe QoS parameters. SLAs between ISPs and home users are usually referred to as Terms of Service, whose primary

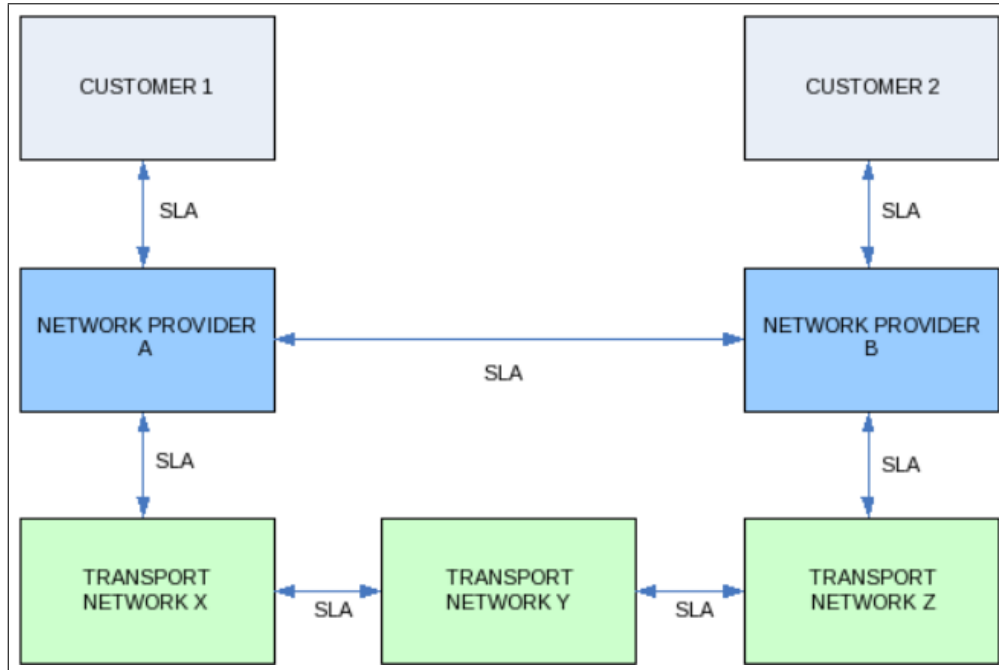


Figure 3.1: Internet business model.

concern is to limit ISPs responsibility in case of abnormal services, rather than protect end users [3]. The actors involved in the Internet business have very distinct roles, and is quite common that enterprises (from different layers of the business model) form tighter partnership to offer more competitive services.

The TNVE business model is presented in Figure 3.2. First of all, there is a clear distinction between the network infrastructure layer and the AS layer. The network infrastructure layer encompasses L1 carrier networks spread throughout the globe, where any single large geographical region has more than one L1 network that spans its entire area. These carriers are distributed in a mesh-like topology, and exchange data via interconnection lines. In our vision, as stated before, to fulfill future Internet applications requirements, NGNs will need more than just data hauling services from the network infrastructure. For instance, to establish a circuit with QoS guarantees that crosses multiple carrier networks, a complex exchange of control messages must take place between the involved parties. Also, at some point of the circuit setup, some network "entity" with a minimum set of a-priori known TE and monetary cost information must be responsible for determining the sequence of L1 networks to negotiate the circuit. Carriers are not willing to disclose this kind of information, even when the means to publicize it are available.

We introduce the concept of a TNA (*Transport Network Alliance*) to denote a set of

3.2. Internet and TNVE Business models

35

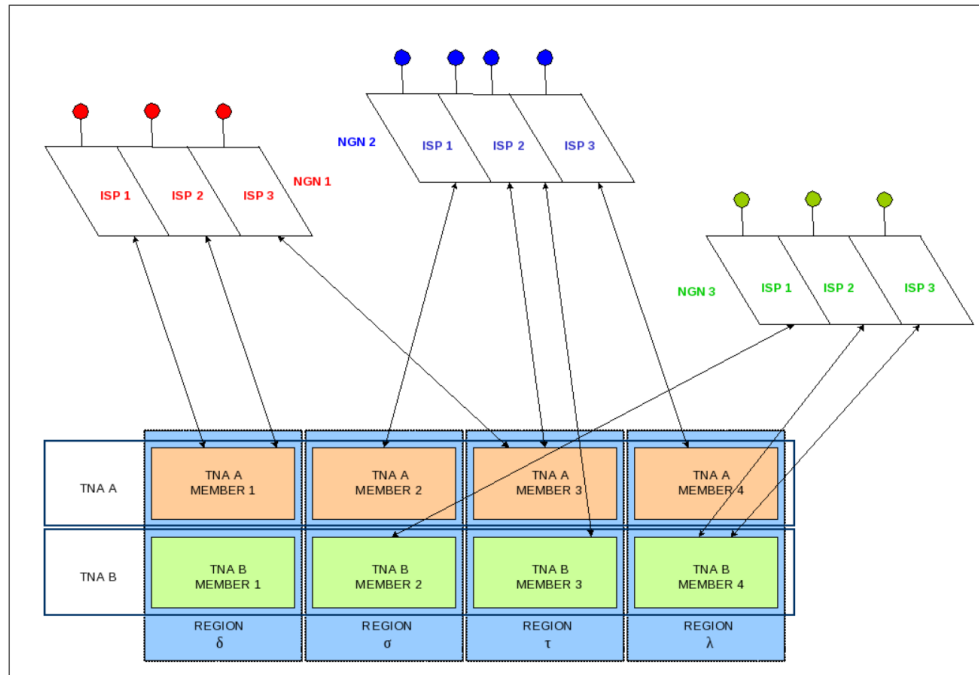


Figure 3.2: TNVE business model.

L1 networks whose members have a stronger business relationship and a higher level of trust between themselves, strictly needed for advanced, multidomain circuit provisioning. The functionalities that enable the management of such circuits are the core of the TNVE architecture, which is described later. Members of a given TNA have complete freedom to operate their L1 networks and have SLAs with non-members TNA carriers. In fact, these interconnections are needed to assure global reachability for traditional (i.e., nowadays) services. A TNA can be seen as a "carrier of carriers", and clients are free to choose one or another TNA in function of their service portfolio and prices.

The AS layer embraces a collection of NGN planes, each of them specialized in supporting specific advanced services and running a dedicated stack of protocols. Each NGN plane is composed of interconnected ISPs using NGN-specific technologies. ISPs of a given NGN plane can request on-demand infrastructure services to the TNAs (usually 1, no more than 2) which they are attached to. With an on-demand infrastructure, ISPs can easily grow in the same pace as traffic grows, drastically reducing costs associated with network planning, and making cheaper and less risky the deployment of advanced services.

Customers attached to ISPs can be completely unaware of the TNA presence. For service-specific traffic, customers of ISPs of the same NGN plane can

transparently rely on TNA services. For "legacy" services, the IP traffic between clients of different ISPs and NGN planes, attached to different TNA members, traverses L1 networks the way it does today (considering that both NGNs provide IP compatibility).

3.3 Application Scenario

An alliance of providers could have a large, variable number of members, with different (non GMPLS) and not fully compatible L1 intradomain CPs, yet with the ability of DP interconnection. Moreover, NGNs will require more than "just" protected point-to-point circuits or a L1VPN with QoS guarantees [80]. NGNs will require from the underlay a client manageable, on demand virtual infrastructure, fully customized with different types of connections (P2MP or MP2P) that is completely isolated from other instances running on top of the same alliance. To accomplish that, a management plane to coordinate alliance members is not suitable. A fully distributed alliance CP is needed to provide all the requested services, so as to maximize the resource utilization of the members of a given TNA in a transparent fashion. In fact, the TNVE has a fully distributed CP that spans the boundaries of the TNA members, which are independent domains capable of providing at least point-to-point circuits with QoS guarantees and are interconnected with other TNA members via dedicated lines. Hence, the TNVE must cope with the diversity of interdomain CPs in a TNA. Another challenging goal of the TNVE is not only to provide customized and isolated virtual infrastructures to NGNs, hiding all the complexities of interdomain connections, but also to offer services that TNA members, by themselves, are unable to provide, e.g., data compression or non-point-to-point circuits. Thus, the TNVE active functionalities supersede by far the management of TNA federated resources, where end services are limited to a composition of individual network capabilities.

The services offered by the TNVE are the following:

- unidirectional or bidirectional point-to-point circuits and unidirectional P2MP/MP2P circuits, with assured bandwidth and delay upper bounds. Circuits can be requested one by one or in a batch (i.e., according to a predefined traffic matrix);
- survivability. Three classes of recovery strategies can be assigned to point-to-point links: 1+1 protection, pre-planned and pre-configured restoration, and post failure, best effort restoration;
- on-the-fly cryptography and data compression (also multimedia coding/decoding). Life span of legacy applications can be extended,

3.4. FIE Architecture

37

and the implementation of new ones is simpler, saving costs. Actually, there are already available commercial devices that perform cryptography on-the-fly at line card speed;

- AAA (*Authentication, Authorization and Accounting*). Client NGNs must have unique digital credentials to securely access a pre-established set of services. Also, a precise tracking of TNVE services used by individual NGNEs (*NGN Elements*) is available;
- peer reachability. Each NGNE have access to a list of other NGNEs that are reachable through the TNVE, even if none of them are connected. This can be very useful for service discovery at the NGN layer.

To provide all these services, the TNVE must know which are the capabilities of each TNA member, the monetary costs related to resource allocation, and the TNA topology, i.e., how TNA members are interconnected and which is the state of the interconnection links. When a service is requested, the TNVE uses this information to decide which TNA members will be involved in the service provisioning, in order to guarantee the requested QoS, while also minimizing network resources utilization, monetary costs and service disruption probability. Then, the TNVE must perform the advanced active part of the service (if requested) and coordinate all the involved TNA members to effectively provide the end-to-end service. All these tasks are conducted by the FIE (*Flexible Infrastructure Element*). In addition, the FIE is also responsible for offering an access interface to the NGNEs. Therefore, FIEs must be embedded or attached to TNA members border devices. The FIE architecture is described, in more detail, in Section 3.4.

The TNVE scenario is depicted in Figure 3.3. NGNEs of multiple NGNs are attached to FIEs. Each NGNE can requests to its corresponding FIE different types of services, offered in a complete transparent and isolated way, which means that neither the NGNEs are aware of the complexity of the physical layer, nor the transport infrastructure is shared by multiple NGNs.

3.4 FIE Architecture

To enable automatic provisioning of interdomain circuits across GMPLS-controlled networks, the IETF approach aims at extending the capabilities of the intradomain CP, creating "interdomain aware" versions of the GMPLS protocols. Therefore, the establishment of a circuit that traverses a number of domains is accomplished through a direct negotiation among all the CPs entities of the involved domains. To avoid some pitfalls of this "single layer" solution and facilitate interdomain provisioning, a *Service Plane* was proposed

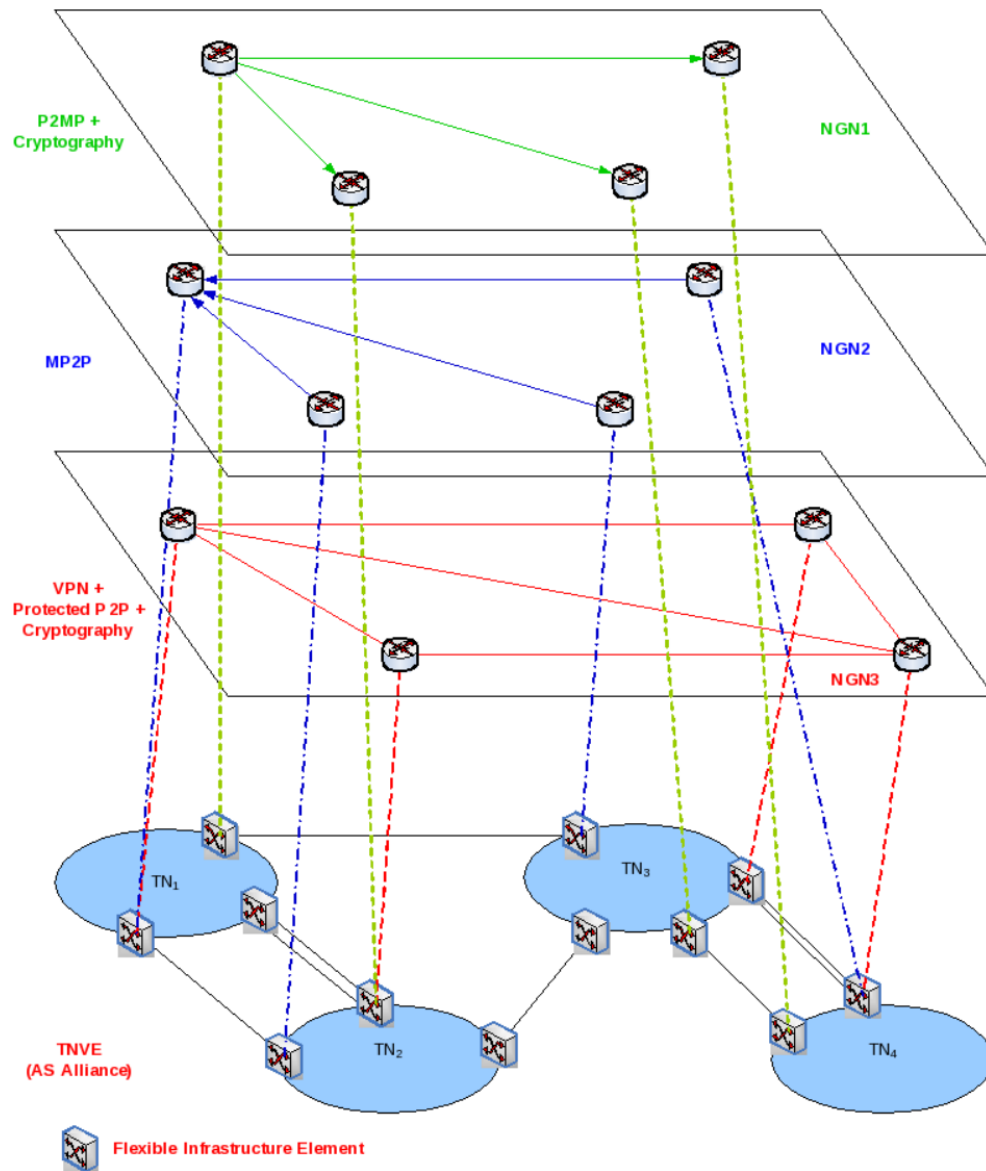


Figure 3.3: TNVE scenario.

3.4. FIE Architecture

39

in [16]. The Service Plane is on top of the set of GMPLS CPs of the involved transport networks, performing the interdomain portion of the path computation, signalling and routing. This concept of introducing a new, upper layer is reused by the TNVE to achieve its ambitious goals in a heterogeneous TNA environment. However, to envision a complete virtualized L1 infrastructure to provide support to NGNs, the concepts introduced by the Service Plane are not enough to realize it.

As stated previously, all TNVE tasks are carried out by the FIEs, that are attached to the border network elements of the TNA members'. The set of FIEs composes an overlay that globally controls the TNA members via their local, intradomain CPs, while performing exclusive active services. All of this on behalf of the NGNs on top of it. FIEs manage their resources (TNA members and interconnection links) with ad hoc protocols to realize functions like automatic discovery, advertising and allocation of resources. These tasks are already well performed by standard GMPLS intradomain protocols. As the overlay formed by the FIEs can be considered as a single control domain, standard intradomain GMPLS protocols can be adapted to perform TNA control functions without the need to develop new protocols from scratch. FIEs use modified versions of GMPLS link management, routing and signalling protocols to satisfy the TNVE needs. The state machine of the customized protocols remains the same, just the control data format is changed to describe TNA resources. To exchange information between them, FIEs can use residual bandwidth in the intradomain control channels and interconnection links. As a last resort option, complete out-of-band links could be used.

The FIE architecture is composed of a number of building blocks that perform CP (including virtualization control) and DP functions, plus SSEMs (*Service Specific Elaboration Modules*) to realize data manipulation. The FIE architecture is shown in Figure 3.4. The functions of each FIE building block are described below:

VC (*Virtualization Control*)

This entity is responsible for satisfying the infrastructural needs of client NGNs in a customized and scalable fashion. Each NGNE has its own infrastructure management interface to request specific services, whose access is granted upon a secure credential validation. Also, a virtual infrastructure control engine (to transparently access the TNA resources), a detailed peer tracking and a service accounting systems are designated to each NGNE. All these per-NGNE elements compose a slice space. Slice spaces are managed by the VC in a complete isolated manner, in order to guarantee robustness and privacy.

The Slice Space diagram is presented in Figure 3.5.

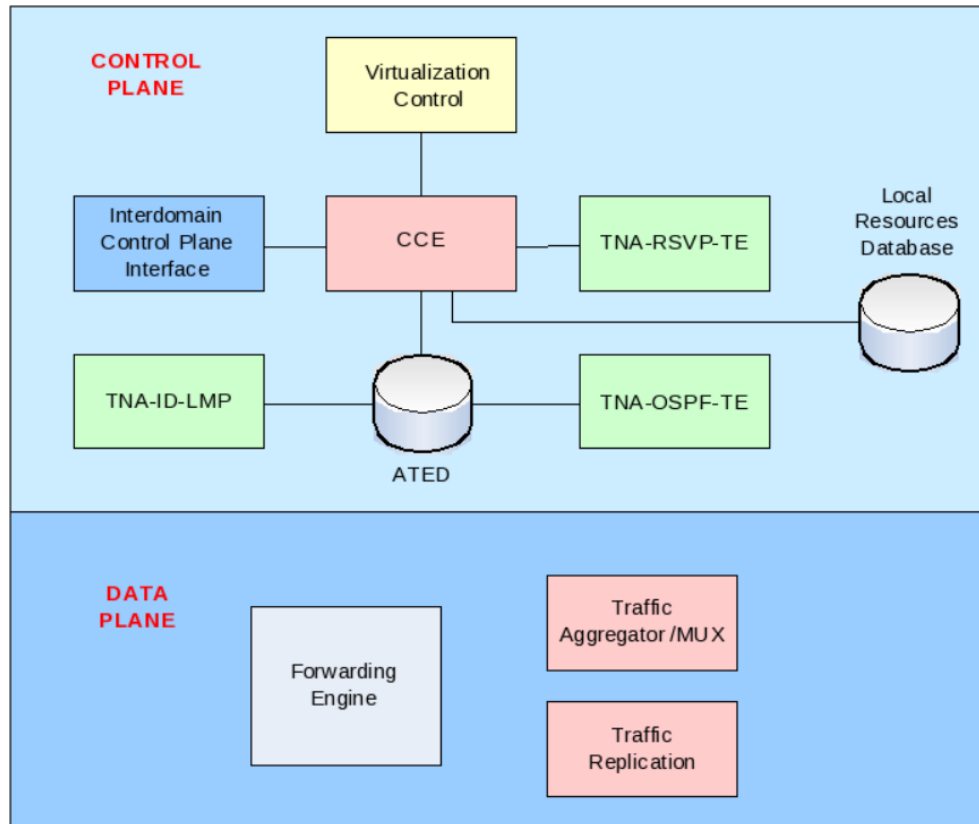


Figure 3.4: FIE architecture.

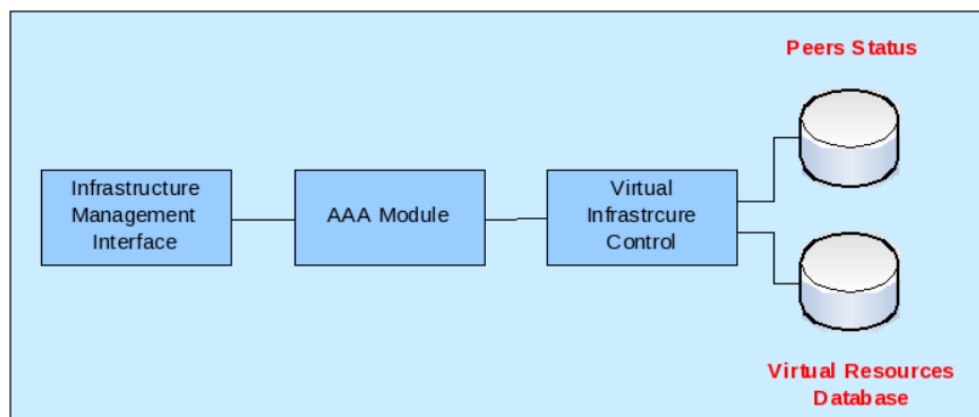


Figure 3.5: A slice space.

3.4. FIE Architecture

41

CCE (*Chain Computation Element*)

When a circuit setup request arrives from the VC, the CCE engine calculates the domain chain and the interconnection links that will be used for the primary and backup (if requested) circuits. This computation uses as input the TNA topology information retrieved from the ATED (*Alliance Traffic Engineering Database*), which essentially describes the capabilities and monetary costs of TNA members and interconnection links. When the computation is finished, the CCE contacts the intradomain CP (an ordinary PCE, for instance) of the TNA member, which the FIE is attached to setup the local sub-path. Also, the CCE triggers signalling messages to other FIEs that are attached to the border elements indicated by the domain chain, to proceed with the setup of all sub-paths via local intradomain CP. In case of P2MP or MP2P circuits, branch FIEs must setup more than one local sub-path, and traffic replication or aggregation must be carried out by DP entities. These mechanisms will be further discussed later. When the signalling protocol reports that the interdomain path is ready to be used, the VC is informed and the path properties are inserted in the Local Resources Database (LRD). The LRD describes all the end-to-end circuits whose ingress element is the FIE itself, including the local resources associated with that service, and its utilization rate. A sub-utilized point-to-point circuit can be used to carry data from other NGNE, through multiplexing or grooming techniques, depending on the transport technologies.

Intradomain CP Interface

All communications between the FIE and the intradomain CP entities of the TNA members is made possible by this interface. It is responsible for forwarding (or even translating) the sub-path setup and teardown requests (from the CCE and the VC, respectively) to the intradomain CP. Also, it receives (or captures) signaling messages from the local CP entities regarding intradomain circuits (like successful establishments or unrecoverable failures). These messages can trigger FIEs recovery procedures or the teardown of the whole interdomain circuit, for example.

TNA-OSPF-TE

This module uses the OSPF reliable flooding mechanism to disseminate throughout the TNVE all the information necessary to compute the domain chain. FIEs also rely on TNA-OSPF-TE to achieve neighbor discovery, in the same way OSPF routers do. Additionally, this module is responsible for propagating the list of NGNEs that are using local FIE services, used by the peer tracking systems in the slice spaces. The information is carried by opaque LSAs in the form of TLV triplets, and no modification to the OSPF state machine is required. Three new top TLVs are defined, along with their sub TLVs:

1. TNA Member

- TNA ID (unique ID number);
- FIE ID (unique ID number inside a TNA);
- Circuit bandwidth (list);
- Maximum delay;
- Monetary cost (per circuit).

2. Interconnection link

- Local FIE ID (unique ID number);
- Remote FIE ID (unique ID number);
- ID (local number);
- Available bandwidth;
- Maximum delay;
- Flags (span protection for now).

3. Client NGNE

- TNA ID (unique ID number);
- FIE ID (unique ID number inside a TNA);
- NGN ID (unique ID number);
- NGNE specific address (list of addresses from NGN address space).

There is not a standard procedure to measure or estimate the maximum delay of data carried by circuits inside a TNA member. Each TNA member is free to define its numeric valor in function of propagation, processing, emission and differential delays. More than it, the maximum delay express the "willingness" of a TNA member to receive circuit setup requests from other members and use its resources as part of interdomain paths, to the detriment of intradomain paths. Of course, TNA members will have to honor the publicize maximum delays or penalties must be applied.

TNA-RSVP-TE

This module uses the RSVP-TE signaling protocol to allow FIEs to manage and recover interdomain circuits. When a circuit setup is requested, RSVP Path messages carrying an ERO object describe the sequence of FIEs the path will traverse. Depending on the ID of the next "hop", each FIE knows if an intradomain sub-path must be requested or if an interconnection link must be used to stitch the intradomain sub-paths. TNA-RSVP-TE also implements the extensions to support P2MP and MP2P circuits.

TNA-ID-LMP

A lightweight version of the LMP protocol is used to correlate

3.4. FIE Architecture

43

interconnection link properties between adjacent FIEs, from different TNA members. The acquired information is propagated by the TNA-OSPF-TE.

Forwarding Engine

NGNs can have a packet format completely different from IP, and may be unsuitable to be directly carried by TNA members. The forwarding engine must adequate (e.g. by encapsulation) NGNs traffic to be transported by the TNVE. This can be realized using a thin adaptation layer, for example Ethernet. NGNs can also request active services, like on-the-fly cryptography or compression. In this case, the incoming traffic from NGNs must be forwarded to the SSEM before entering the long-haul circuit. In the case of P2MP and MP2P circuits, branch FIEs must act as traffic aggregators or replicators. These scenarios are depicted in Figure 3.6 and Figure 3.7, respectively. Incoming data could have to be converted to the electronic domain (in case of optical transmission) to be replicated or multiplexed, and then readapted to the output interfaces. These are the roles of the Traffic Replicator and Aggregator modules.

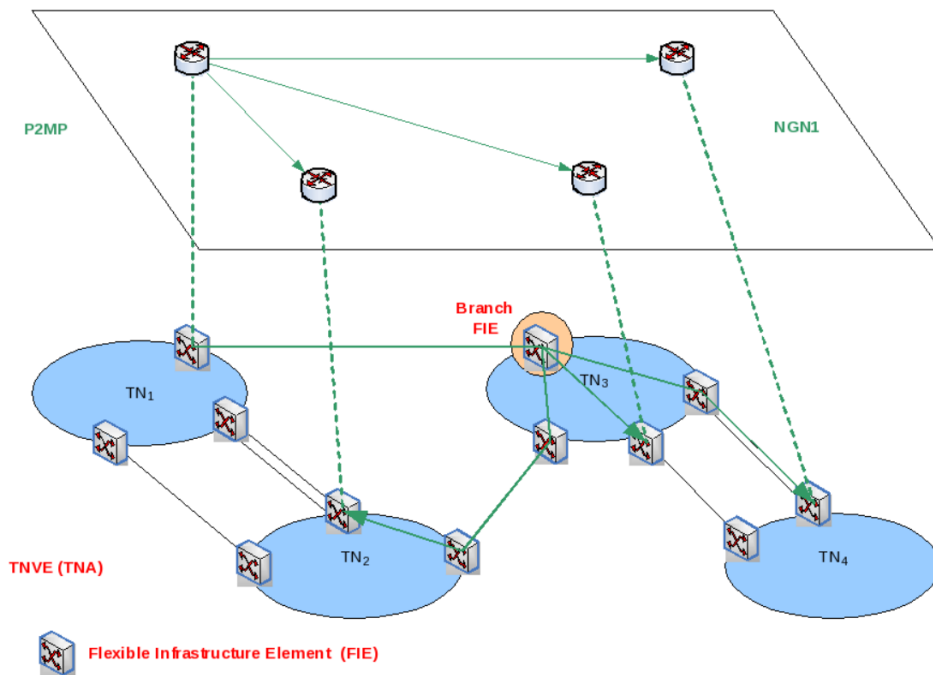


Figure 3.6: P2MP circuit provisioning.

SSEM

This is a collection of hardware devices capable of transforming traffic

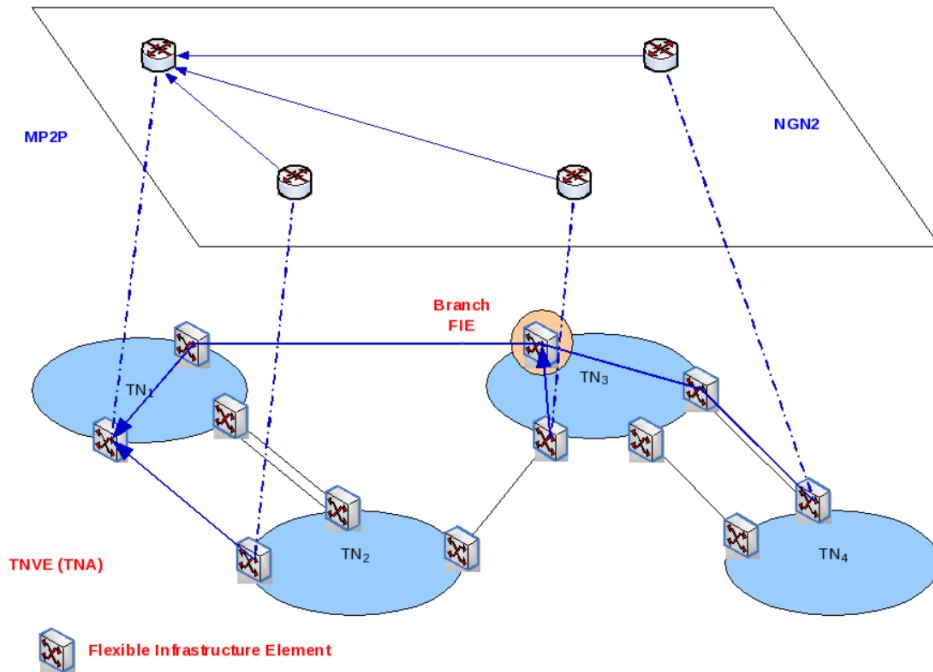


Figure 3.7: MP2P circuit provisioning.

payload at interface speed. Nowadays there are already available commercial devices capable of encrypting and compressing data on-the-fly, and new features are expected in the near future, e.g., virus scanning. New modules can be added to SSEM, providing rapid deployment of new services for client NGNs.

3.5 Case Study

To better demonstrate the TNVE operations, a step-by-step P2MP unidirectional circuit setup is detailed using the scenario illustrated in Figure 3.8 and Table 3.1, Table 3.2 and Table 3.3.

In that scenario, NGN A uses the infrastructure service provided by TNA T, that is composed of four carriers: TN1, TN2, TN3 and TN4. NGN A has 6 NGNEs associated with FIEs distributed across the TNA. The associations between NGNEs and FIEs are shown in Table 3.1. Table 3.2 describes the circuit capabilities and prices of the TNA members, while Table 3.3 depicts the TE properties of interconnection links. These three tables compose the ATED and the Virtual Resources Database, that is made available to every FIE by the TNA-OSPF-TE protocol. The sequence of steps in order to establish a P2MP

3.5. Case Study

45

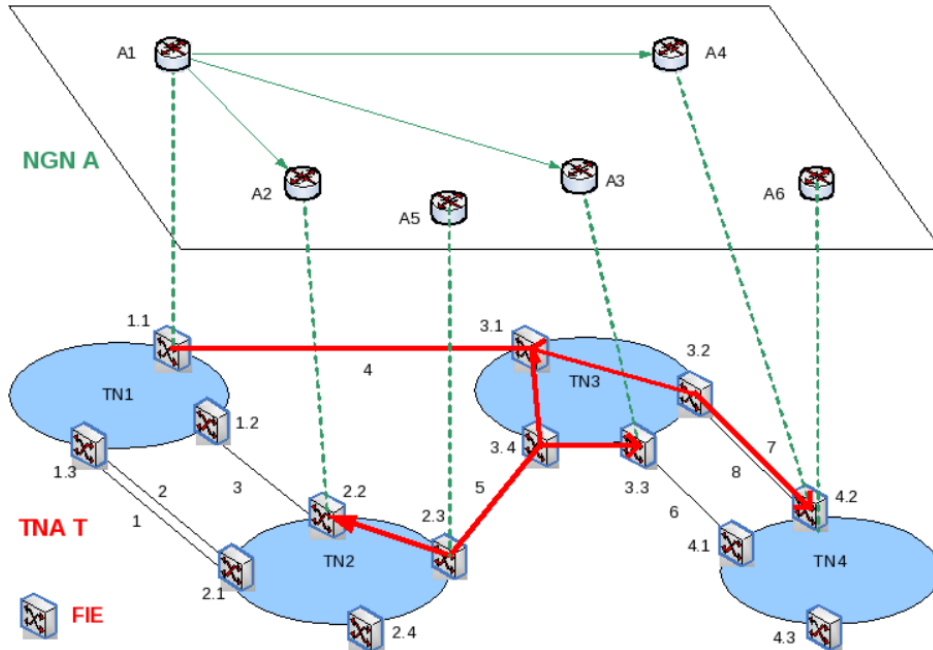


Figure 3.8: Case study scenario.

Table 3.1: Association between NGNEs and FIEs.

FIE ID	NGNE ID
1.1	A1
2.2	A2
2.3	A5
3.3	A3
4.2	A4,A6

circuit from NGNE A1 to A2, A3 and A4 is described below:

1. NGNE A1 requests to its associated FIE (FIE 1.1) to setup a P2MP circuit to A2, A3 and A4 with the following TE metrics: bandwidth = 2.5 Gb/s and maximum delay ≤ 80 ms.
2. After receiving and validating the circuit setup request from A1, FIE 1.1 must know which FIEs will be the end-points of the circuit. This is accomplished retrieving the information from Table 3.1.
3. The CCE at FIE 1.1 calculates the P2MP path from FIE 1.1 to FIEs 2.2, 3.3

Table 3.2: TNA Members Characteristics.

TNA Members	Bandwidth (Gb/s)	Max. Delay (ms)	Monetary Cost
TN1	1,2.5,10	5	1600
TN2	2.5,10,40	2	2000
TN3	2.5,10,40	5	2000
TN4	1,2.5,10	8	1500

Table 3.3: Interconnection Links TE Properties.

Link ID	Available Bandwidth (Gb/s)	Max. Delay (ms)
1	1024	10
2	1024	20
3	2048	25
4	4096	20
5	4096	10
6	1024	20
7	1024	20
8	2048	25

and 4.2, using the information described in Table 3.2 and Table 3.3. The calculated path is shown in Figure 3.8.

4. FIE 1.1 reserves the resources in link 4 and sends a RSVP-TE PATH message to the next FIE in the path (FIE 3.1). The PATH message contains the sequence of FIEs and interconnection links that must be used to setup the circuit.
5. FIE 3.1 will act as a branch node, and upon receiving the PATH message from FIE 1.1, asks to the local intradomain PCE (using its Intradomain CP interface) to establish two point-to-point unidirectional interdomain paths from itself to FIE 3.2 and to FIE 3.4. FIE 3.1 can wait or not for the successful establishment of the interdomains paths to forward the PATH message, in order to minimize crankback probability or setup delay.
6. FIE 3.2 receives the PATH message from FIE 3.1. It reserves the resources in link 7 and forwards the PATH message to FIE 4.2
7. FIE 3.4 receives the PATH message from FIE 3.1. It also will act as a branch

3.5. Case Study

47

node, and asks to the local intradomain PCE to setup a point-to-point circuit from itself to FIE 3.3. Also, it reserves the resources in link 5 and forwards the PATH message to FIEs 3.3 and 2.3.

8. FIE 2.3 receives the PATH message from FIE 3.4. It asks to the local intradomain PCE to setup a circuit from itself to FIE 2.2 and forwards the PATH message to FIE 2.2
9. FIE 4.2 receives the PATH message from FIE 3.2. It informs NGNE A4 that the P2MP circuit is being established, and sends a RESV message back to FIE 3.2.
10. FIE 3.3 receives the PATH message from FIE 3.4. It informs NGNE A3 that the P2MP circuit is being established, and sends a RESV message back to FIE 3.4.
11. FIE 2.2 receives the PATH message from FIE 2.3. It informs NGNE A2 that the P2MP circuit is being established, and sends a RESV message back to FIE 2.3.
12. FIE 3.2 receives the RESV message from FIE 4.2 and forward it to FIE 3.1. Also, it commit the resources in link 7. Table 3.3 will be locally updated, and TNA-OSPF-TE will disseminate the new available bandwidth of link 7.
13. FIE 2.3 receives the RESV message from FIE 2.2 and forward it to FIE 3.4.
14. FIE 3.4 receives the RESV messages from FIE 2.3 and 3.3, and send them to FIE 3.1. Again, it can wait to receive all RESV messages and send them in a batch, or send them as they arrive. Also, it commit the resources in link 5.
15. FIE 3.4 receives the RESV messages from FIE 3.2 and 3.4, and forward them to FIE 1.1.
16. Once FIE 1.1 receives all RESV messages, it commit the resources in link 4, make a entry in the Local Resources Database for the new circuit, update the accounting information and informs NGNE A1 that the circuit is ready to be used.

Chapter 4

Open Framework for Service Software Routers

This chapter presents a DS-MPLS framework, entirely developed within this thesis work using exclusively open software, that enables the cross-connection of DS domains using both *DS over MPLS* and *DS-aware MPLS TE* models. The first section contextualizes the need of open frameworks for routers and briefly describes the application scenario. Section two presents an overview of works related to the proposed open framework. Next, section three introduces the DS-MPLS framework architecture and its main features. Finally the chapter concludes with a description of two *live distributions* of the framework.

4.1 Preamble

Simulators were - and still are - one of the most commonly used tools to aid the design of network protocols and other entities like schedulers, filters, classifiers, and so on. Simulators were usually used as just part of the development process of network device software, before a prototype code was specifically written for the given platform. As PCs became more and more powerful and cheaper, and the public availability of open source software grew, PCs started to be used not only as simulator platforms, but also as experimental, fully capable router prototypes. When playing such a role, PCs are called SRs (*Software Routers*) [81].

Traditionally, SRs are based upon open source, Unix-like OSes (*Operating Systems*), such as GNU/Linux and BSD variants. These systems offer complete solutions to filter and manipulate layers 2 and above PDUs, with a level of flexibility which is comparable only with costly, enterprise-level systems. Moreover, SRs can offer programmability features, allowing the development of third party extensions, completely new functionalities and fine-tuning of existing ones. Routers belonging to this category are called FSRs (*Flexible Software Routers*).

In the last years, SRs are playing an important role even in the market. Leading network vendors are exploring the SOHO (*Small Office/Home Office*) market with embedded SRs, using mixed open and close source solutions (for the OS and protocol stack, respectively). More recently, SRs are being used as platforms to the development of NGNs and networking learning tools. In addition, it is expected that SR deployment will continue to grow.

Despite the current popularity of SRs, the successful deployment of SR-based solutions is far from straightforward, when considering research and other academic purposes. On the one hand, off-the-shelf SRs, despite being based on commodity hardware and open source OSES, are shipped with close networking stacks that limit (or even prevent) the development of new capabilities. On the other hand, complete open SRs quite often are developed to evaluate single advances in network components, like optimizations for an existing routing protocol or a brand new scheduler. These enhancements tend to be ad-hoc, intrusive hacks in the original source code, which make its maintenance, reuse and integration with other projects a complex and time-consuming task. Even when a number of complementing network functionalities are provided by an open SR solution, there is a clear lack of operating and orchestrating procedures to benefit from their cooperation. QoS support is a concrete example of this deficiency. There is a great number of open source tools to enforce QoS, but few guidelines are available on how to combine them to produce specific routing behavior.

These factors have driven the development of an open source framework for FSRs, called DS-MPLS open framework. It effectively combines DS and MPLS, still the most advanced IP-based QoS and TE architectures, allowing fine-grained flow control. Moreover, it considers the integration of external-developed modules to bring intelligence to the network [8]. These third-party modules can perform service-specific duties like deep packet inspection or cryptography, to cite a few examples. However, the unique features of the DS-MPLS framework are the high levels of integration between components and service automation, without compromising flexibility for further extensions.

4.2 Related Works

During the last decade, a number of projects have contributed to the characterization of SRs. Some of these projects are discussed below, as well as current works regarding router programmability.

TEQUILA

TEQUILA (*Traffic Engineering for Quality of Service in the Internet*, at

4.2. Related Works

51

Large Scale) [82] was a thirty-month European Union funded project, that started January 1st, 2000. Its goal was to specify and develop protocols and mechanisms for negotiating, monitoring and enforcing service level specifications, within the DS model. One of its main contribution was the development of an open source RVSP-TE daemon [83].

BORA-BORA

The 2005 project BORA-BORA (*Building Open Router Architectures Based On Router Aggregation*) [84] was a two years program funded by the Italian Ministry of Education, University and Research. It focused not only on software components, but also on hardware subsystems to develop scalable and reliable architectures, aiming to the definition of models for the dimensioning, the implementation of distributed functionalities, and the study of load balancing schemes among interconnected routers.

Quagga

Quagga [85] is the leading open source protocol suite for Unix platforms. It provides implementations for RIP, OSPF and BGP. The OSPF implementation offers support for Opaque LSAs and also partially implements TE extensions. Quagga is shipped with many Linux distributions and is an integral part of the majority of commercially available SRs.

XORP

XORP [86] is an open, modular networking platform. It supports RIP, OSPF, BGP, PIM-SM (*Protocol Independent Multicast - Sparse Mode*) [87], IGMP (*Internet Group Management Protocol*) [88], MLD (*Multicast Listener Discovery*) [89], VRRP (*Virtual Router Redundancy Protocol*) [90], and others. Despite the lack of TE extensions, XORP is under constant development. An extensive API is available to add router programability.

Mikrotik RouterOS

RouterOS [91] is a closed source Linux-based SR, designed to run in commodity PCs and also in custom boards made also by Mikrotik. Parts of the Linux kernel (essentially drivers) and almost all userspace networking tools are proprietary. It offers a wide range of applications, but no flexibility at all.

Vyatta Core

Vyatta Core [92] is a freely available, open source Linux-based SR, designed to run in commodity PCs and also in other general purpose architectures. Enterprise-class management and security tools are available (also source code) in subscription editions. In 2008, the XORP-based routing engine was substituted with Quagga.

Netkit

Netkit [93] is a SR based on the Debian Linux distribution [94], developed by Roma Tre University Computer Networks Laboratory [95]. It supports MPLS tunnels and the XORP suite of protocols. Virtual machines running Netkit are supported via UML (*User-Mode Linux*) [96], and virtual networks can be created using an XML-based (*eXtensible Markup Language*) language known as NetML [97]. Netkit is primary used to perform network experiments as a learning tool.

Openflow

Openflow [98] aims to enhance network elements with a more advanced forwarding mechanism, using a centralized programmability approach. Instead of traditional layer 2 and 3 forwarding based on destination addresses, openflow-enable devices rely on a 10-tuple *flow header* to forward data, which is composed by fields as source and destination MAC and IP addresses and TCP ports. Entries are added and removed in the *flow table* by centralized entities called *controllers*, which can be policy driven. This way, complex VLANs and even pseudo circuits can be managed in a openflow cloud. Openflow is currently available for Linux-based SRs and the NetFPGA[99] platform. A number of vendors are already embedding openflow in their devices.

Junos SDK and Cisco AXP

The Achilles heel of high-end networking apparatuses is flexibility. Pushed by the need to add intelligence to the network core, programability capacity is, together with cloud virtualization, one of the current top concerns of vendors. The Junos SDK [100] and Cisco AXP [101] bring to routers the capability of running third party code developed in Unix-like environments, using standard languages as C, C++, Perl, Python and Java. Custom API and libraries are made available to also enable the interaction of applications with parts of the control and management planes of routers.

4.3 Framework Architecture and Features

In a network composed of DS-FSRs developed in accordance with the DS-MPLS Open Framework, the following features are available:

- automatic provisioning of uni- and bidirectional circuits, specifying absolute values for bandwidth, end-to-end delay and monetary cost metrics;
- two distinct modes of LSP provisioning - a simpler, centralized mode and a distributed, robust mode;

4.3. Framework Architecture and Features

53

- specialized LSPs, such as E-LSPs, L-LSPs and LSPs for carrying CT[0-8];
- PCE with a variety of path computation algorithms to be chosen;
- MAM and RDM bandwidth allocation models support;
- TED and LSP topologies described in XML language, for easy integration with upper layer systems;
- comprehensive traffic statistics;
- ingress shaping support [102];
- advanced multifield classifier;
- service-specific modules support.

The main building blocks that together form the architecture of the DS-MPLS Open Framework are depicted in Figure 4.1, and detailed as follows:

NMS (*Network Management System*)

The NMS provides graphical user and command-line interfaces to allow the administration of network resources. NMS users can setup and teardown LSPs, manage FEC-LSP associations, and analyse the operational status of DS-LSRs, links, protocols and virtual circuits. Currently, only the PCC is implemented.

TED

The TED describes the properties and state of links and DS-LSRs. It is defined using XML.

LSP-DB (*LSP Database*)

Details the characteristics of installed LSPs. As TED, it is also defined using XML.

PCE

The PCE is responsible for processing LSP requests originated from the PCC embedded in the NMS. In case of bidirectional circuit requests, the PCE computes a path that is able to accommodate two LSPs in opposite directions. After the path computation phase, LSPs are established in a centralized or distributed manner.

LCS (*LSR Control System*)

The LCS configures all routers involved in a LSP setup or teardown. When signaled from the PCE, it uses a secure, centralized approach to contact and configure DS-LSRs.

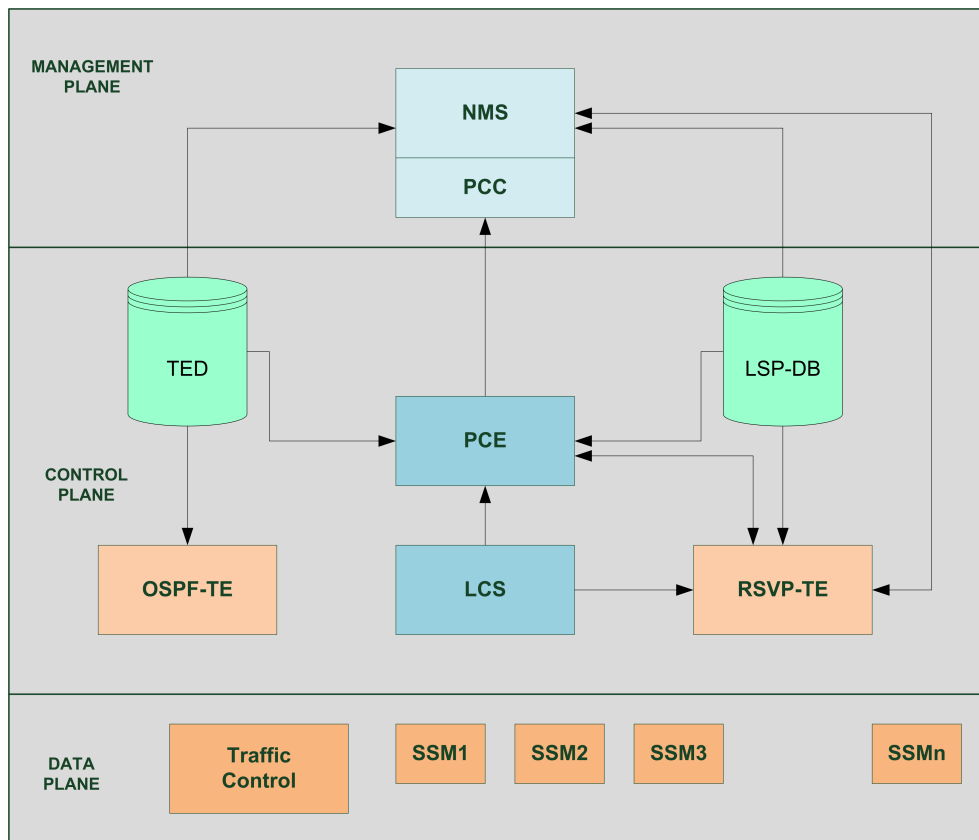


Figure 4.1: DS-MPLS Open Framework architecture.

RSVP-TE

The RSVP-TE fully distributed signaling mechanism is used to perform a robust, automatic provisioning of LSPs.

OSPF-TE

Responsible for maintaining, updating and synchronizing the TED across the DS-LSRs.

Traffic Control

Enforces the QoS of data flows. Includes all DS entities like classifiers and schedulers in order to perform traffic policing. Is also responsible to perform MPLS switching.

SSM (*Service-Specific module*)

SSMs are external programs developed to bring intelligence to the network. This way, other than just forward packets, DS-LSRs can elaborate data,

4.3. Framework Architecture and Features

55

simplifying the creation of upper layer applications and overlays. SSMs can perform basic services such as cryptography and compression, as also more advanced features such as video transcoding. A number of SSMs are being envisaged for future projects.

The next subsections describe the data and control planes, detailing the manual and automatic LSP management procedures.

4.3.1 Data Plane

The DS-MPLS Open Framework DP uses both standard GNU/Linux routing and traffic control functionalities, as well as a number of unofficial tools to add new capabilities, such as label-switching routing. Most of the times these external add-ons required intrusive kernel patching and hacking of userspace tools. To end up with a full-feature DP on a single system, custom patches were developed to circumvent version mismatches. Moreover, the DP integration required the utilization of applications with unusual, poorly documented features, which lead to obscure bugs and unexpected behaviors. Indeed, some functionalities were successful applied on a trial and error basis, before being polished and refined. Although part of the DP features are presented in other projects previously introduced (Section 4.2), none of them includes all features of the presented DP. However, what really differentiates the proposed framework is the fact that, while other projects limit themselves to just ship software, the framework provides a concise integration of networking tools to enable even greater capabilities. The resulting DP and its features are elucidated below. For a working example of manual configuration of three LSPs in a DS-LSRs, see Appendix A.

LSP establishment

MPLS support in the DS-MPLS open framework is granted by the MPLS for Linux project [103]. It consists of several patches in the Linux kernel and in the following userspace tools: `iproute2` (traffic control in Linux) [104], `iptables` (layer 3 packet filtering) [105], and `ebtables` (layer 2 cell/frame filtering) [106]. Also, a new tool called `mpls` is introduced to manage LSPs.

To establish E-LSP and L-LSPs, the proposed framework takes advantage of a `mpls` tool feature that allows setting the EXP field of the shim header and the *TC index* field of the packet buffer descriptor (internal to the FSR) in function of the DSCP of an IP packet. The TC index plays an invaluable role when assigning MPLS to schedulers, which is detailed later. While the mapping between DSCPs and the EXP field is standard for L-LSPs, the classic DSCP-EXP mapping (Table 4.1) is used by the framework when establishing an E-LSP:

In order to setup a LSP using the framework, the following steps are mandatory:

Table 4.1: Association between DSCP and EXP fields used to setup E-LSPs

DSCP	EXP
0x00 (BE)	0
0x2E (EF)	1
0x0A (AF 11)	2
0x0C (AF 12)	3
0x12 (AF 21)	4
0x14 (AF 22)	5
0x1A (AF 31)	6
0x1C (AF 32)	7

- On ingress and core DS-LSRs, the creation of an entry in the NHLFE (*Next Hop Label Forwarding Entry*) table, with the respective translations from DSCP to EXP and TC index;
- On core and egress DS-LSRs, the creation of an entry in the ILM (*Incoming Label Map*) table;
- On core DS-LSRs, the cross-connection between ILM and NHLFE entries.

The code listing 4.1 shows a typical E-LSP configuration on a core DS-LSR.

```

mpls ilm add label gen 12110 labelspace 0
NHLFE_CMD=mpls nhlfe add key 0 instructions \
    ds2exp 0x3f 0x00 0 0x2e 1 0x0a 2 0x0c 3 \
    0x12 4 0x14 5 0x1a 6 0x1c 7 \
    exp2tc 1 0x01 2 0x02 3 0x03 4 \
    0x04 5 0x05 6 0x06 7 0x07 \
    push gen 13011 nexthop eth1 ipv4 172.16.13.2`
NHLFE_KEY=echo $NHLFE_CMD | awk '{print $4}'`
mpls xc add ilm_label gen 12110 ilm_labelspace 0 \
    nhlfe_key $NHLFE_KEY

```

Listing 4.1: Typical E-LSP configuration on a core DS-LSR

When creating manually an LSP (i.e., without CP intervention), these commands must be typed directly in the console of every DS-LSR belonging to the path. It is a time consuming, error pruning, repetitive task.

4.3. Framework Architecture and Features

57

FEC to LSP Binding

The proposed framework provides a great amount of flexibility regarding FEC definitions. Almost every field of layers 2 and 3 packet headers, as well as part of the layer 4 header, can be used to define a FEC. Due to limitations of userspace tools, FECs can not be directly associated with LSPs. The standard GNU/Linux networking was not designed with MPLS support in mind. Thus, it is not straightforward to assign to different LSPs (that could be configured on different interfaces) two packets with exactly the same source and destination IP addresses. To overcome this problem, the framework uses multiple routing tables with higher lookup priorities, when compared to the default, BE routing table. By using *forward marks*, each FEC is associated with a specific routing table, which usually contains a single entry that is related to a specific LSP (in fact, its NHLFE). This solution permits the finest QoS routing granularity, although it is also the most expensive in terms of resources. However, the framework allows the use of multiple NHLFE entries per table.

The FEC-NHLFE association can be summarized in the steps below:

1. incoming packets belonging to a FEC are tagged with a specific *forward mark*;
2. during the routing phase, the forward mark of a packet is used to locate the specific routing table that contains the NHLFE entries associated with its FEC;
3. when more than one NHLFE entry is present, the IP longest-prefix match is used to finally choose the correct LSP to route the packet

The following code samples exemplify how FEC-LSP bindings are performed by the framework. Both examples use the data flows and NHLFEs shown in Table 4.2. In code listing 4.2, each routing table contains exactly one entry, while code listing 4.3 allows tables to have two or more entries.

Table 4.2: Example of association between FECs and LSPs

Flow	QoS profile	Specs	NHLFE	IF
1	voice	DSCP = EF	40	eth3
2	video stream	DSCP = AF11, dst. IP = 74.5.31.8	23	eth1
3	video stream	DSCP = AF11, dst. IP = 89.16.4.4	31	eth0

```

iptables -t mangle -A PREROUTING \
            -m dscp --dscp-class EF \
            -j MARK --set-mark 1
iptables -t mangle -A PREROUTING -d 74.5.31.8/32 \
            -m dscp --dscp-class AF11 \
            -j MARK --set-mark 2
iptables -t mangle -A PREROUTING -d 89.16.4.4/32 \
            -m dscp --dscp-class AF11 \
            -j MARK --set-mark 3
ip rule add fwmark 1 table 1 prio 30001
ip rule add fwmark 2 table 2 prio 30002
ip rule add fwmark 3 table 3 prio 30003
ip route add default via 172.16.12.1 dev eth3 mpls 40 table 1
ip route add default via 172.16.18.6 dev eth1 mpls 23 table 2
ip route add default via 172.16.12.1 dev eth0 mpls 31 table 3
    
```

Listing 4.2: FEC-NHLFE binding, with only one NHLFE entry per table

```

iptables -t mangle -A PREROUTING \
            -m dscp --dscp-class EF \
            -j MARK --set-mark 1
iptables -t mangle -A PREROUTING \
            -m dscp --dscp-class AF11 \
            -j MARK --set-mark 2
ip rule add fwmark 1 table 1 prio 30001
ip rule add fwmark 2 table 2 prio 30002
ip route add default via 172.16.12.1 \
            dev eth3 mpls 40 table 1
ip route add 74.5.31.8/32 via 172.16.18.6 \
            dev eth1 mpls 23 table 2
ip route add 89.16.4.4/32 via 172.16.12.1 \
            dev eth0 mpls 31 table 2
    
```

Listing 4.3: FEC-NHLFE binding, with more than one NHLFE entry per table

While three tables and forward marks are used in code listing 4.2, only two are used in code listing 4.3. However, if a fourth flow is added, and it is assigned to a fourth LSP and destined to 74.5.31.8 or 89.16.4.4, another table with a new forward mark must be configured. That happens because, whenever two routes with identical prefixes are presented in the same table, the first one is always used, leaving the other one pointless.

Hierarchical Scheduling

QoS enforcement is provided by the DS-MPLS open framework through a complex tree of hierarchical packet schedulers, available on every interface of DS-LSRs. The hierarchical scheduler tree is illustrated in Figure 4.2.

4.3. Framework Architecture and Features

59

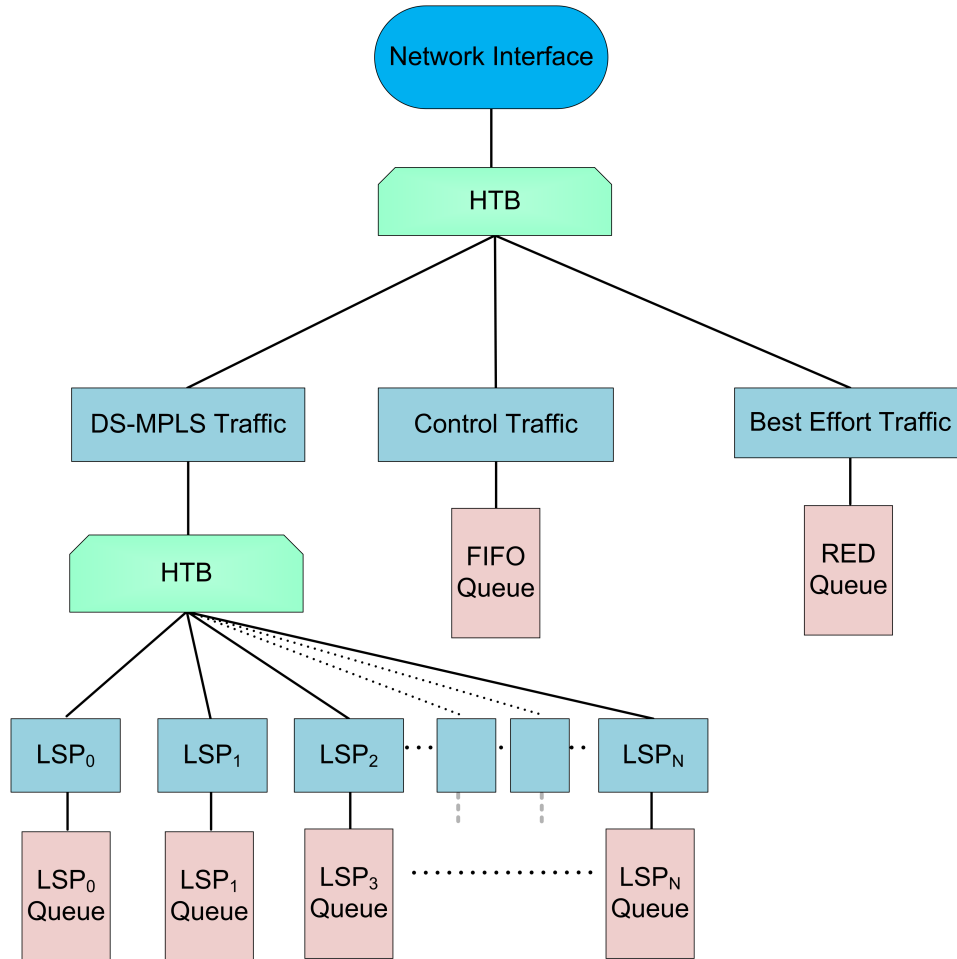


Figure 4.2: Hierarchical scheduler tree.

The reservable bandwidth for interfaces is initially divided between three groups of traffic: control traffic, DS-MPLS traffic and BE traffic. By reserving a fraction of bandwidth to control traffic, the CP is isolated from the DP (out-of-band approach). Also, by reserving a small amount of bandwidth to BE traffic, minimal service levels are guaranteed, avoiding complete starvation of the lowest packet class. To promote the bandwidth separation, the HTB (*Hierarchical Token Bucket*) [107] packet scheduler is used. While the interface bandwidth quota reserved to control traffic is managed by a FIFO queue, the BE traffic is subject to a RED queue. The DS-MPLS traffic bandwidth share is managed by another HTB scheduler, which is responsible to guarantee the nominal bandwidth of LSPs.

When establishing a new LSP, a specific hierarchical scheduler subtree is

configured according to the type of the LSP. The predefined scheduler subtrees for E-LSPs, L-LSPs carrying AF traffic and L-LSPs carrying EF traffic are depicted in Figure 4.3 and Figure 4.4.

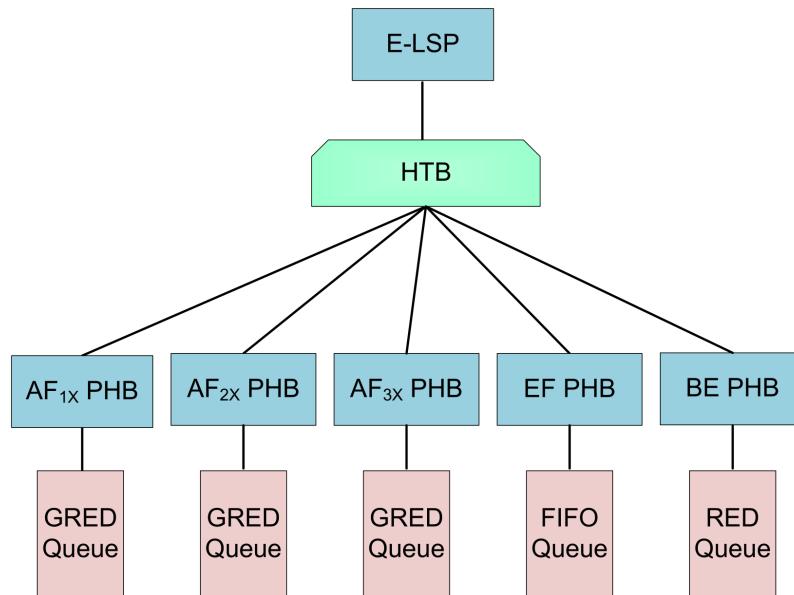


Figure 4.3: E-LSP scheduler subtree.

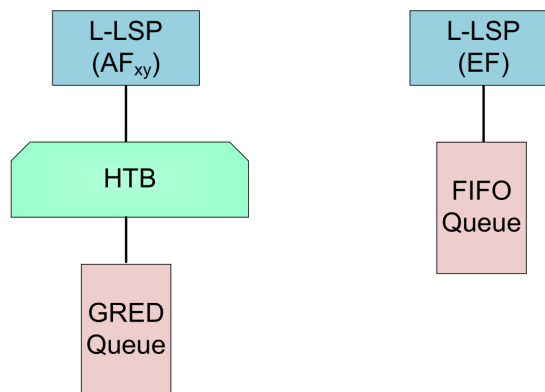


Figure 4.4: L-LSP scheduler subtrees for AF and EF traffics.

The E-LSP subtree divides the LSP bandwidth among the eight PHBs as seen in Table 4.1. Each one of the three AF classes can use up to 100% of the LSP bandwidth, with a guaranteed bandwidth equal to 25% of the LSP capacity. To EF

4.3. Framework Architecture and Features

61

PHB, a guaranteed level of 15% of total bandwidth is offered, with no borrowing allowed from other classes. There is no bandwidth reservation for the BE traffic inside LSPs. The drop precedence for the AF PHBs are enforced using the GRED (*Generalized Random Early Detection*) scheduler, a multiple virtual queues RED variant. GRED uses the TC index (which is directly related to the EXP shim header field) to discriminate AF packets. The EF packets are subjected to a small buffer FIFO queue, BE packets are managed by a RED queue.

L-LSP scheduler subtrees are simpler. For L-LSPs carrying AF traffic, a GRED scheduler is attached to it. In case of EF traffic, a small FIFO queue is used.

Every outgoing Ethernet frame carrying a shim-header is initially designated to the DS-MPLS traffic portion of interfaces. Later, each frame must actually be assigned to the subtree that correctly corresponds to the LSP to which it belongs. This is accomplished using a specially crafted *TC filter* rule (tc is one of the main iproute2 tools), that uses the label field of the shim header to queue the packet in the appropriate scheduler subtree. The code listing 4.4 exemplifies a TC filter rule:

```
tc filter add dev eth1 parent 11:0 protocol 0x8847 pref 9 \
    u32 match u32 0x042d6000 0xffffffff000 \
    at 0 flowid 11:2
```

Listing 4.4: Label matching TC filter rule

where `eth1` refers to an interface, `11:0` is the handle to the DS-MPLS traffic portion of the interface, `0x8847` is the Ethernet type code for MPLS packet, `9` refers to the filter rule priority, `0x042d6000` and `0xffffffff000` refer to the label and bitmask respectively, `0` indicates the bit offset and finally `11:2` indicates the handle for the LSP scheduler subtree. More details can be found in Appendix A.

4.3.2 Control Plane

The open framework CP provides all the means necessary to allow the NMS to automatically setup and teardown LSPs. The central entity of the CP is the PCE, which directly communicates with the PCC integrated in the NMS. Currently, six constraint-based path computation algorithms are implemented in the PCE. The link metrics used by these algorithms are the reservable bandwidth, the TE cost, the delay and the monetary cost. The TED and the MPLS protocols were extended to support the non-standard metrics. An example of a TED described in XML language is presented in Appendix B.

A LSP setup request must contain the following information: source and destination nodes, the required bandwidth, and the type of LSP to be established. Optionally, it can be also specified the maximum delay, the monetary cost and the path algorithm to be used. Moreover, when a bandwidth allocation model

is used (MAM or RDM), it is also necessary to specify the priority and the preemption levels for the LSP. Moreover, when a bidirectional circuit is requested, the PCE computes two LSPs with the same path and properties, but with opposite directions. The list below describes all the possible LSP types to be chosen:

- Non specified (default, usually used for tunneling)
- E-LSP
- L-LSP carrying Default Forwarding (BE) traffic
- L-LSP carrying AF1x traffic
- L-LSP carrying AF2x traffic
- L-LSP carrying AF3x traffic
- L-LSP carrying AF4x traffic
- L-LSP carrying EF traffic
- LSP carrying TE-Class0 traffic (CT[0-7] + preemption priority[0-7])
- LSP carrying TE-Class1 traffic (CT[0-7] + preemption priority[0-7])
- LSP carrying TE-Class2 traffic (CT[0-7] + preemption priority[0-7])
- LSP carrying TE-Class3 traffic (CT[0-7] + preemption priority[0-7])
- LSP carrying TE-Class4 traffic (CT[0-7] + preemption priority[0-7])
- LSP carrying TE-Class5 traffic (CT[0-7] + preemption priority[0-7])
- LSP carrying TE-Class6 traffic (CT[0-7] + preemption priority[0-7])
- LSP carrying TE-Class7 traffic (CT[0-7] + preemption priority[0-7])

A LSP teardown request must specify only the identifier of the LSP to be removed. In case of a centralized process (i.e., without using the fully distributed MPLS routing and signaling protocols), the PCE entity is also responsible for dealing with LSP teardown requests. Considering that the PCE implemented in the framework already has interfaces with the TED, LSP-DB and LCS, it was a natural design decision. Also considering the centralized LSP provisioning process, the PCE relies on the LCS to enforce the configuration of the DS-LSRs involved. The interaction between the NMS, PCE and LCS while processing setup and teardown requests are depicted in Figure 4.5 and Figure 4.6, respectively. Diversely from the other components of the CP that were implemented in C, the LCS was prototyped with the `bash` language. The LCS is in fact formed by two

4.3. Framework Architecture and Features

63

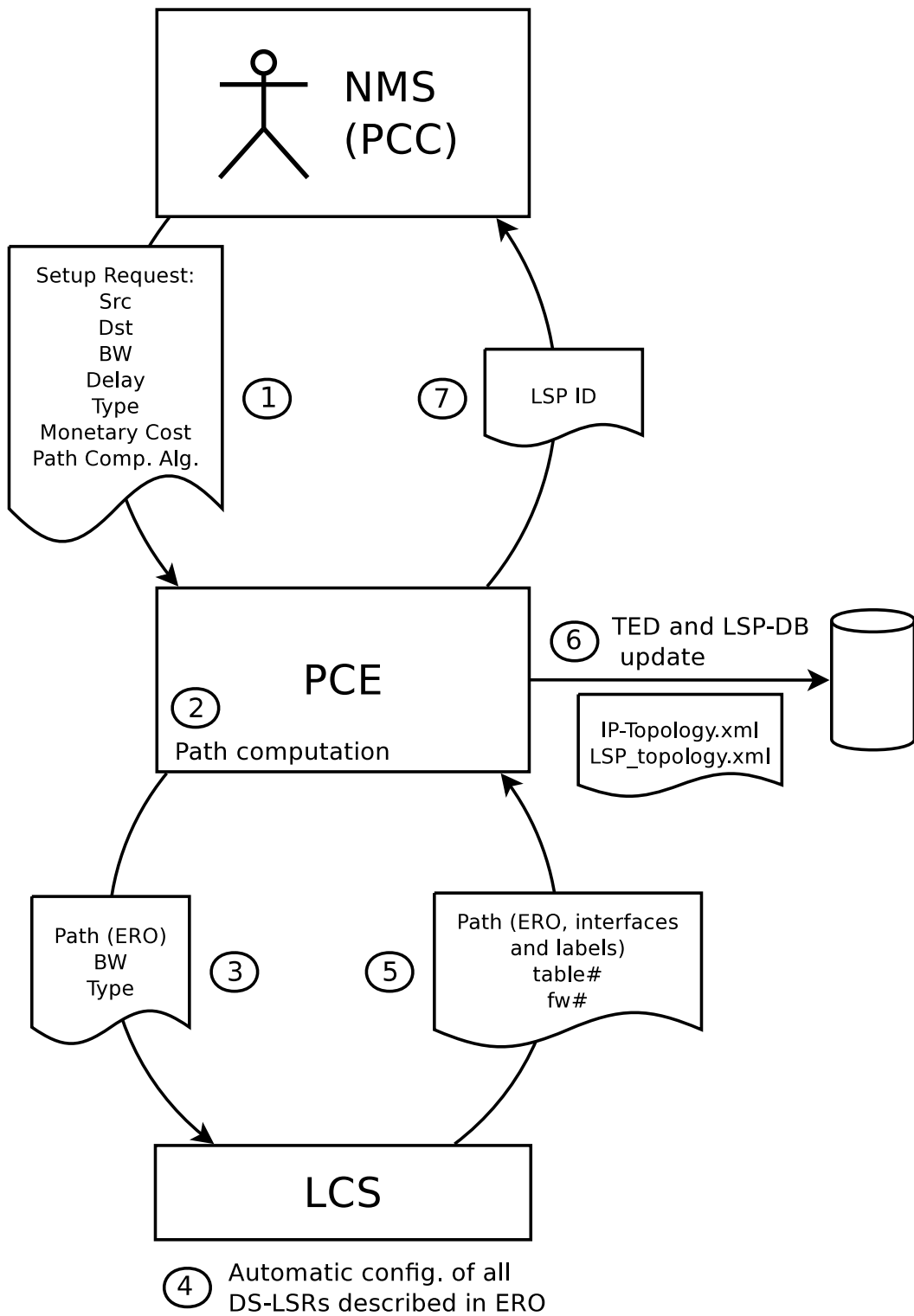


Figure 4.5: Collaboration diagram for the centralized LSP setup procedure.

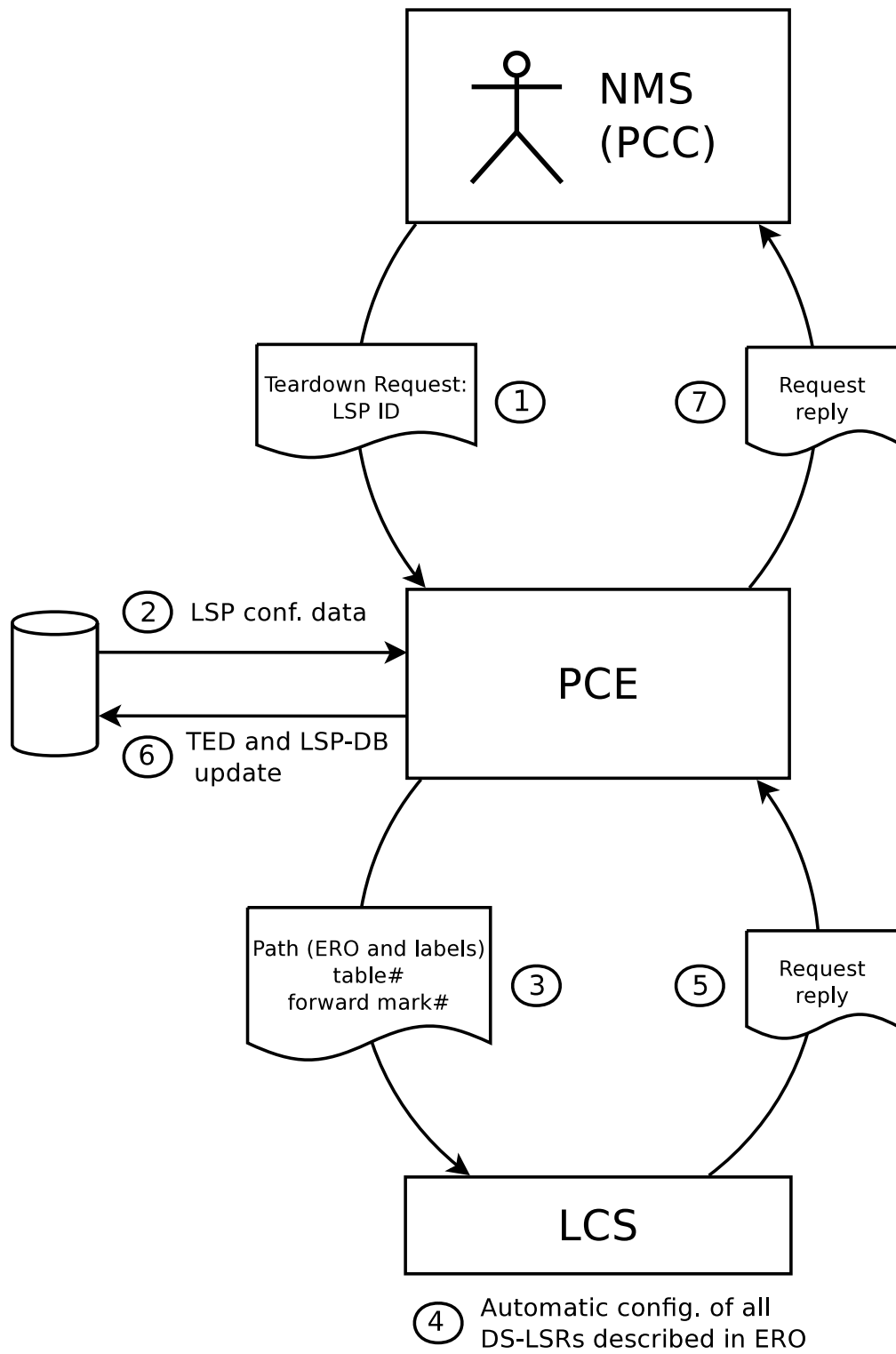


Figure 4.6: Collaboration diagram for the centralized LSP teardown procedure.

4.3. Framework Architecture and Features

65

distinct parts: the LCA (*LSR Control Agent*) and the LLEA (*LSR Local Enforcement Agent*). The LCA component is always in the same node as the PCE, no matter if the node is a DS-LSR or a host external to the network. It is responsible to receive the circuit information from the PCE and translate it into configuration commands for all the routers in the path. These commands are the same ones that are used to manually configure the DPs of DS-LSRs. The LCA controls all the DS-LSR specific information such as the labels to be configured in the interfaces. Once the set of commands that each router must execute to configure the LSP(s) are defined, the LCA contacts all LLEAs entities in every DS-LSR that composes the path. Each LLEA is responsible to execute locally the set of commands to configure the LSP(s), and then to report back to the LCA the status of the operation. Figure 4.7 shows the LCS internal components and the interaction with the PCE.

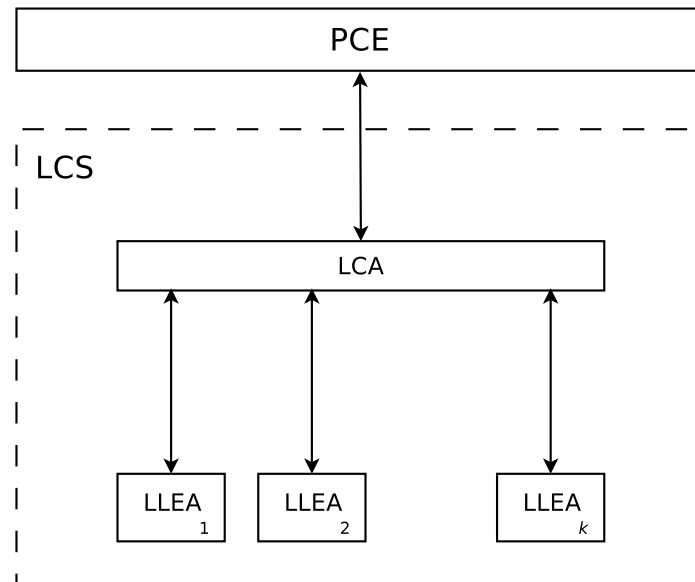
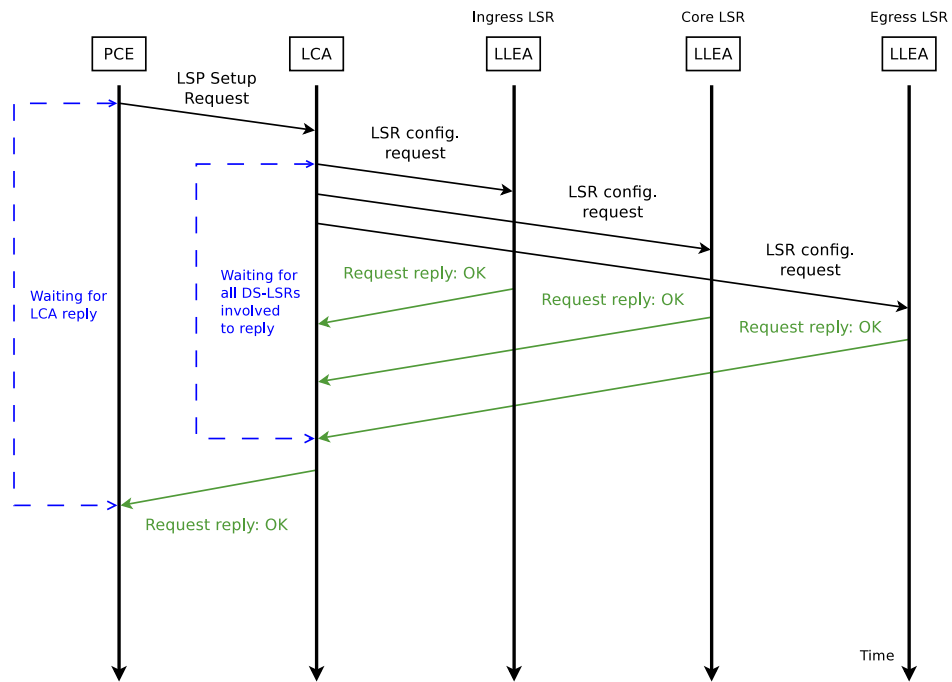
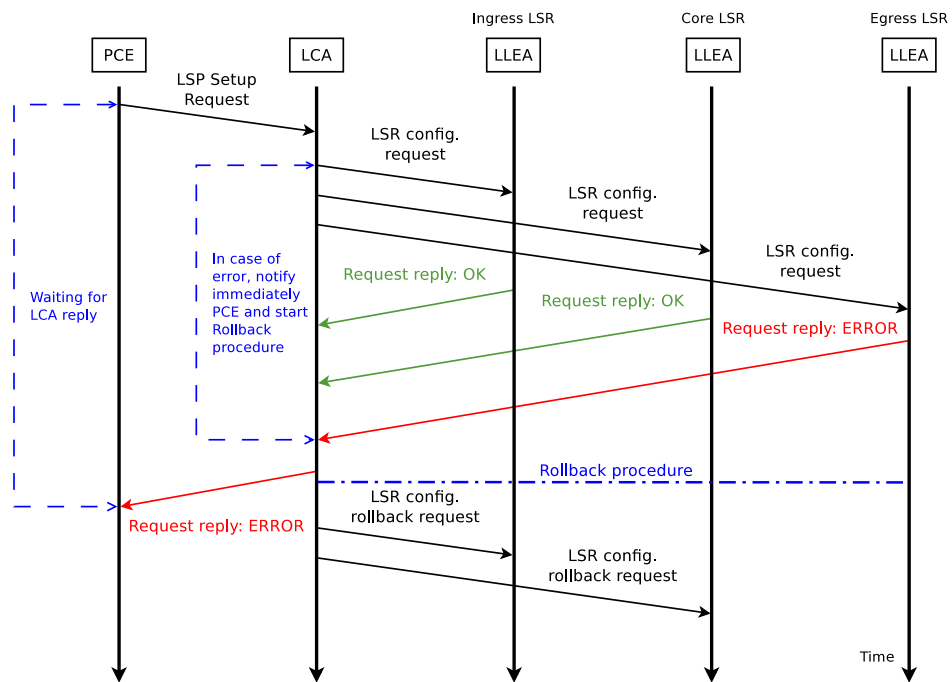


Figure 4.7: LCS internal components and the PCE interaction.

Figure 4.8 and Figure 4.9 detail the sequence diagrams for successful and failed LSP setup and teardown requests. The LSP path is formed by three routers, namely ingress, core and egress LSR. The successful LSP setup and teardown sequence diagrams are almost identical. The LCA receives the information from the PCE, processes it, contacts all LLEAs, blocks its execution until all LLEAs have successfully replied, and finally returns the positive result to the PCE. The failed LSP setup sequence diagram depicts a different behavior. When the egress router reports that the configuration has failed, the LCA, upon receiving the error message from the corresponding LLEA, immediately tells the PCE that the LSP



(a) Successful LSP setup request



(b) Failed LSP setup request

Figure 4.8: LSP setup request sequence diagrams.

4.3. Framework Architecture and Features

67

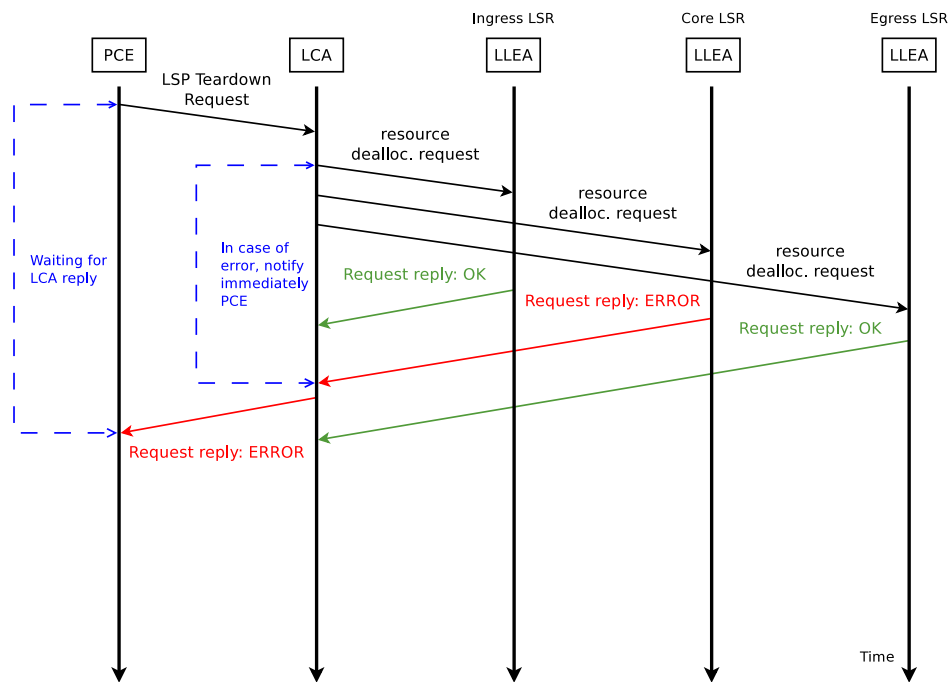
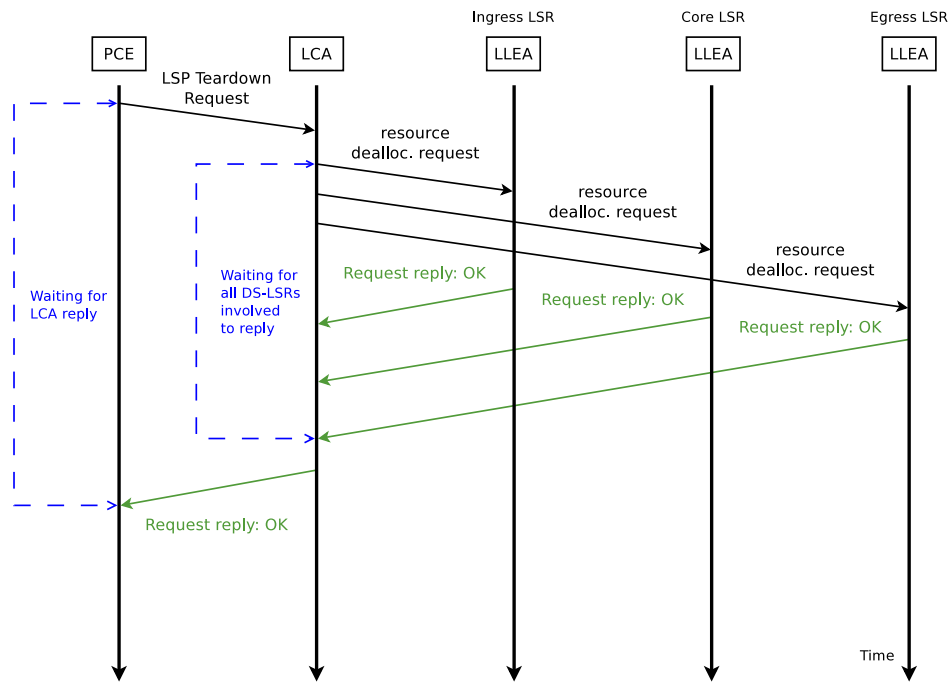
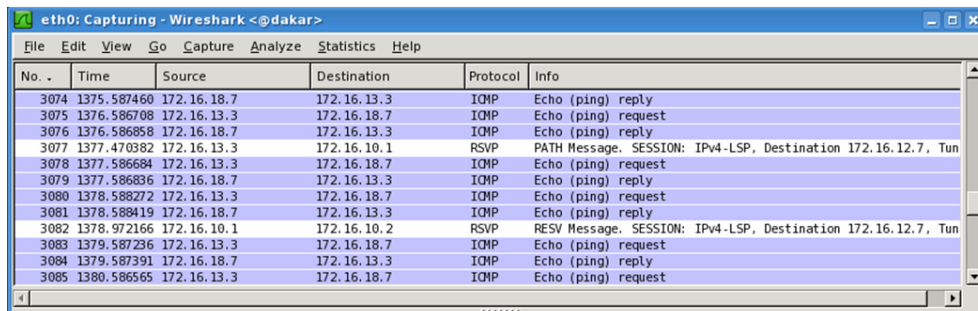


Figure 4.9: LSP teardown request sequence diagrams.

setup has failed, and starts the rollback procedure in all other routers, in order to restore their original state (i.e., as they were before the setup request). The failed LSP teardown request is slightly different, as there is no rollback procedure. In both failure cases, an extensive log is available to help troubleshooting activities. The automatic LSP provisioning through the use of the MPLS protocols is yet under development. To properly flood TE-LSAs describing TE link properties, the OSPF implementation of the Quagga suite of protocols is being used. This solution is based on a dissemination technique previously discussed in a master thesis [108]. The RSVP-TE daemon from the TEQUILA project was renovated and enhanced to become an integral part of the framework, although not all necessary functionalities have been incorporated yet. A packet trace containing PATH and RESV messages generated with the RSVP-TE daemon is shown in Figure 4.10.



No.	Time	Source	Destination	Protocol	Info
3074	1375.587460	172.16.18.7	172.16.13.3	ICMP	Echo (ping) reply
3075	1376.586708	172.16.13.3	172.16.18.7	ICMP	Echo (ping) request
3076	1376.586858	172.16.18.7	172.16.13.3	ICMP	Echo (ping) reply
3077	1377.470382	172.16.13.3	172.16.10.1	RSVP	PATH Message. SESSION: IPv4-LSP, Destination 172.16.12.7, Tun
3078	1377.586684	172.16.13.3	172.16.18.7	ICMP	Echo (ping) request
3079	1377.586836	172.16.18.7	172.16.13.3	ICMP	Echo (ping) reply
3080	1378.588272	172.16.13.3	172.16.18.7	ICMP	Echo (ping) request
3081	1378.588419	172.16.18.7	172.16.13.3	ICMP	Echo (ping) reply
3082	1378.972166	172.16.10.1	172.16.10.2	RSVP	RESV Message. SESSION: IPv4-LSP, Destination 172.16.12.7, Tun
3083	1379.587236	172.16.13.3	172.16.18.7	ICMP	Echo (ping) request
3084	1379.587391	172.16.18.7	172.16.13.3	ICMP	Echo (ping) reply
3085	1380.586565	172.16.13.3	172.16.18.7	ICMP	Echo (ping) request

Figure 4.10: Packet trace containing PATH and RESV messages.

Regarding the fully distributed scenario, the PCE contacts the RSVP-TE daemon to establish the LSP, while the NMS can directly contact the RSVP-TE to teardown a circuit. All other functionalities follow the standards definitions.

4.4 Open Framework Live Distributions

As already stated, to transform a PC-based Linux box into a fully enabled DS-LSR is far from trivial. The complexity of this task can render the reuse of the open framework impracticable for non-highly-skilled users, or when time is a limited resource. To overcome this limitation, two “live” distributions of the DS-MPLS framework have been created. A DS-MPLS live distribution consists of a special version of the framework installed in a bootable DVD, pendrive or any other removable media. A modern computer or a virtual machine can boot one of the live distributions in around one minute. At the boot screen it is possible to choose a console or a graphical interface, as well as an initial keyboard layout. One of the live distributions boot splash screen is shown in Figure 4.11.

4.4. Open Framework Live Distributions

69

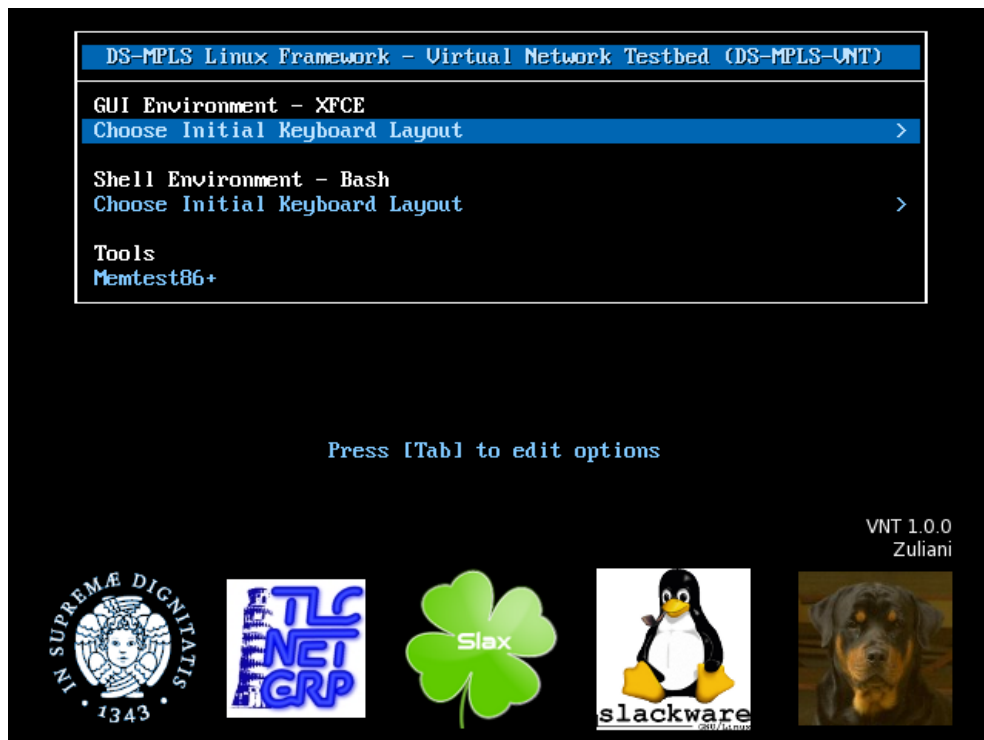


Figure 4.11: One of the DS-MPLS live distribution boot splash screen.

The first live distribution was intended to rapidly add or replace a node in a network. Once the boot process is complete, there is no distinction between a “normal” DS-LSR and one booted with a live distribution. If installed in a writable media such as an pendrive, persistent changes are allowed.

The second live distribution has a different purpose. Instead of transforming a PC into a LSR, the PC is turned into a host of a virtual network, composed of virtual LSRs. UML is used to launch all virtual LSRs and to give them network connectivity. Topologies for the virtual network can be easily created using VNUML (*Virtual Network User-Mode Linux*) [109]. Using a XML file, VNUML allows the specification of nodes, interfaces, virtual links with different capacities and VLANs. A default network topology configuration file is shipped with the live distribution. Therefore, immediately after booting it is possible to launch the virtual network. This operation takes about three minutes to complete in a modern PC.

The default topology is depicted in Figure 4.12. The topology is composed by the host (physical PC) and seven virtual DS-LSRs. The DP is composed by nine links that interconnects all LSRs. Three of them plus the host are interconnected using the “outside” VLAN, that is used to transport traffic to/from the virtual network. If the host has real Internet connectivity, it is shared with the virtual LSRs. In

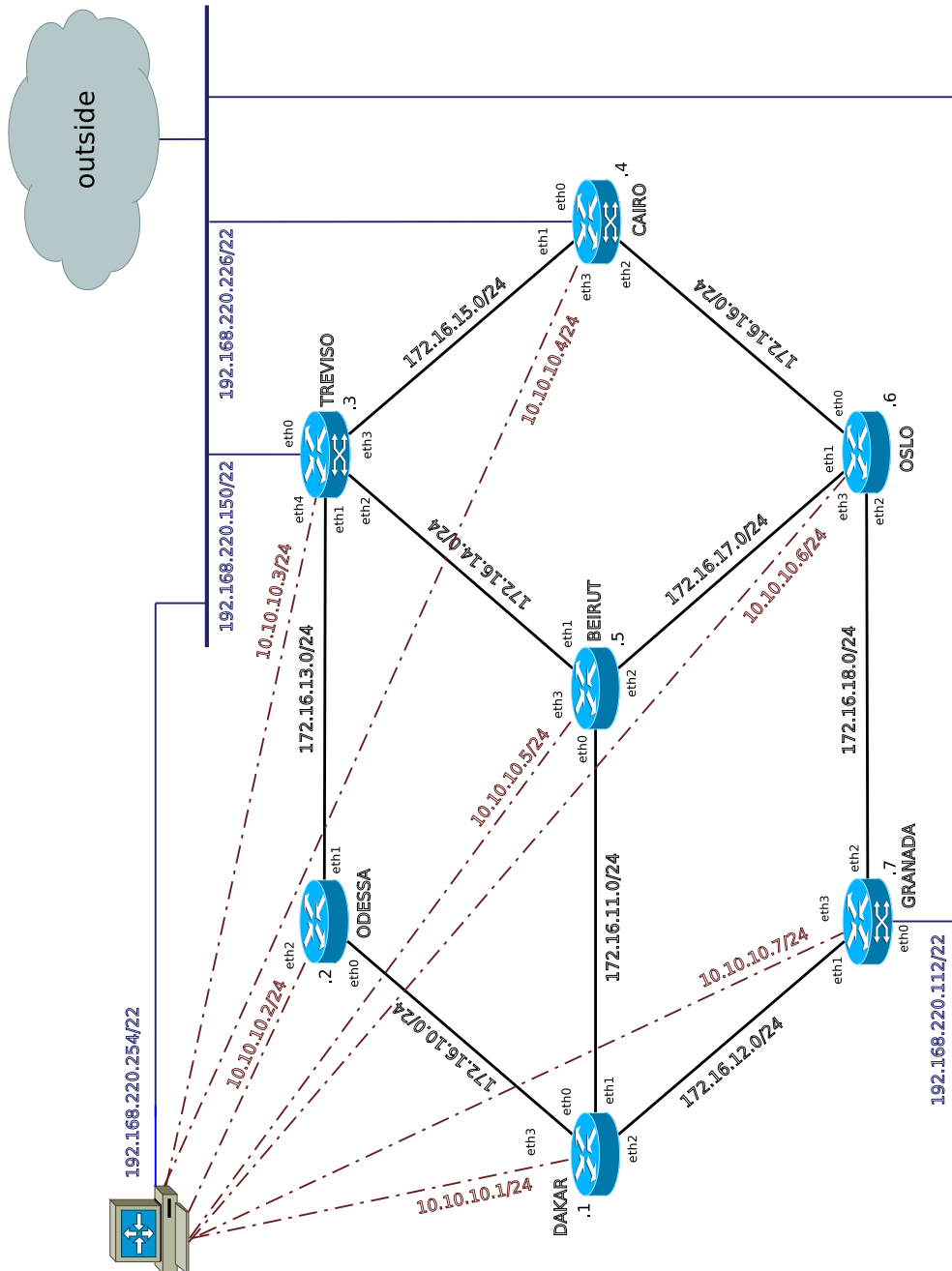


Figure 4.12: TNV topology.

4.4. Open Framework Live Distributions

71

this case, the three LSRs connected with the outside VLAN act as gateways for the others. In addition, a CP VLAN was added. It provides direct connectivity between the host and all the routers. The XML configuration file that enables this topology is detailed in [Appendix C](#).

The ability to have at one’s disposal a fully functional DS-MPLS network, anywhere at anytime, in just minutes, is invaluable. Research simulations and network applications prototyping can be quickly deployed. Moreover, advanced networking teaching can be brought to a completely new level. Practical exercises that today must be conducted inside laboratories with expensive gear or carefully crafted testbeds, can now be carried out in laptops.

Chapter 5

QoT-Assured Survivable LP Provisioning

This chapter presents three novel IA-RWA algorithms specifically envisioned to WRPNs in support of future Internet services. Initially, the state of the art on IA-RWA algorithms is introduced. Next, the proposed algorithms are depicted. Detailed discussions about their mechanisms, including considerations about design decisions, are also presented. The chapter is then concluded with an extensive evaluation of the proposed algorithms, confronting them with variants of IA-RWA algorithms found in the literature.

5.1 State of the Art

WRPNs controlled by an intelligent control plane (such as the GMPLS suite of protocols) are the most promising L1 infrastructure building block to leverage advanced and profitable applications [24, 110]. Despite the RWA problem has been extensively studied in the past decade, the design of RWA algorithms, specifically tailored for WRPNs providing advanced services, poses new challenges yet to be addressed [111, 112]. From the application perspective, these algorithms need to perform online LP provision (there is no information about LP setup requests before their arrival, which usually can not be predicted) as fast as possible to minimize the LP setup delay. They must also guarantee certain levels of QoT and LP survivability, i.e. the capacity to avoid service disruption in case of network failures. From the network perspective, RWA algorithms for WRPNs must optimize the resource allocation and minimize the blocking probability for future LP setup requests. Moreover, they must be enough robust to cope with the high dynamicity of setup and teardown requests for short-duration LPs, while having a low CPU utilization.

As discussed in Section 2.1, optical signals traversing WRPNs are not

regenerated at each hop anymore due to the absence of OEO conversion. Thus, they accumulate transmission impairments that affect the QoT and consequently the end-to-end BER. As WRPNs tend to have meshed topologies with relatively large dimensions, an LP could be unusable due to poor QoT even with plenty of available resources. The wavelength continuity is yet to be considered a hard constraint since all-optical wavelength conversion is still an immature and expensive technology. A number of IA-RWA algorithms have been proposed in the last years, using a variety of network impairment models. However, when LP survivability is also required, the IA-RWA problem becomes much more complex and only a few works are available [113]. In a WRPN where LPs are subjected to transmission impairments, the setup of a new LP can significantly affect the QoT of preestablished LPs. Therefore, resilience mechanisms that are activated after a network failure not only may not guarantee that the traffic being carried will be recovered, but also - and most important - the rearrangement of disrupted LPs can seriously degrade the performance of all previously deployed LPs, even including those not directly affected by the failure [114] [115].

The main goal of all survivable IA-RWA algorithms is to provide LP resilience in case of failures, and usually only a single link failure is considered. The resilience of an LP (called primary or work LP) is achieved by providing another LP (called secondary or backup LP) that is used to deliver the data after the failure. The primary and backup LPs must be link-disjoint (or even node-disjoint as a plus) between themselves. The activation of the backup LP must not affect the remaining active LPs, and the traffic being carried by it should have the same QoS profile as before the failure. Despite all these common characteristics, survivable IA-RWA algorithms can be designed in very different ways, depending on the constraints considered not only for the WRPN itself, but also for the LPs. Survivable IA-RWA algorithms can be classified in function of the network impairment model utilized, the type and levels of resilience offered, the methods used to solve the RWA problem [113].

5.1.1 Network Impairment Models

Transmission in optical fibers is affected by a number of physical impairments. The most relevant ones are the following (as described in Section 2.1: chromatic dispersion, PMD, ASE and nonlinear effects). The predominant impairment depends on many factors, like the quality of fibers and node components, the LP optical signal power and bandwidth, and the wavelength spacing between channels. Network impairment models are used to analytically quantify the influence of transmission impairments on the QoT of LPs. Therefore, it is possible to predict the end-to-end BER of an LP before its deployment. Nowadays, two distinct classes of network impairment models are utilized. To estimate the final BER for a given LP, IA-RWA of both classes rely upon the Q factor, which is a

5.1. State of the Art

75

signal-to-noise ratio. The Q factor is formally elucidated in Subsubsection 5.2.1. Algorithms belonging to the first class consider physical layer impairments as noise-like terms, and the sum of their variances is accounted for the Q factor calculation. The second class of impairment models indirectly deals with impairments, by presetting lower or upper bounds for LP metrics, for instance the LP length in km. This is very useful when the impact of a given impairment in the QoT can not be directly quantified online, as most of the nonlinear effects [113].

5.1.2 Resilience

Basically, LPs resilience techniques can be classified either as pre-configured or pre-planned. In both cases the backup LP is already computed, but only in the former case resources for the backup LP are allocated in advance. More specifically, if the backup LP carries the same traffic as the primary LP, this kind of resilience is called 1+1 dedicated protection. If the backup LP is used for Best Effort traffic (that is preempted on case of failure) or not used at all, it is called 1:1 dedicated protection. Protection is very efficient (service disruption is inferior to 50 ms), but it is also the most expensive kind of resilience [54, 116, 117].

Restoration techniques encompass pre-planned resilience and can be dedicated or shared. In both cases the wavelength remains unused in the fiber links until the restoration mechanisms are activated. Therefore, the fiber remains “dark”, at least for that particular channel. In the case of shared restoration, a wavelength reserved for shared backup remains free to be used for shared backup path computations, i.e., it can (and possibly will) be used to protect more than one LP [116, 117]. Restoration is better for the overall network QoT, because the backup LPs remain dark and do not interfere with the QoT of the primary LPs. Moreover, shared restoration improves the network resources utilization. On the other hand, when a LP must be restored through a pre-planned computation, there is no guarantee that a) it will satisfy the required BER and b) it will not compromise the QoT of other established LPs. The situation is even worse in the case of shared restoration. This happens because, when a new LP must be setup, the IA-RWA engine usually does not take into account the physical impairments of dark wavelengths used to restore LPs. Even if it does, the network status, when the failure occurs, could be completely different with respect to the time the backup LP was computed. This is due to the elapsed time between backup LP computation and primary LP failure that could be really large. Therefore, only by using dedicated protection it is possible to assure absolute QoT for all LPs established in a WRPN in case of failure.

5.1.3 Methods for solving the RWA problem

As stated in [112, 113], RWA algorithms and QoT evaluation processes can be combined in many ways, with different levels of complexity and performance. The best (and most complex) IA-RWA algorithms consider the physical impairments during the RWA phase, and also estimate the end-to-end BER of the candidate LP. As IA-RWA algorithms of this class tend to require longer processing time (that increases the setup delay of LPs), state of the art survivable IA-RWA algorithms [114, 115] take a lightweight approach that divides the RWA problem in two sub-problems. First, the route for the primary LP is calculated using the Dijkstra algorithm (or variations of it). Usually, the cost metric used to weight links is computed in function of one or more physical impairments, i.e., the cost of the link increases as its QoT deteriorates. After the path for the primary LP is calculated, a wavelength assignment heuristic such as FF, LF, BF, RP or MU is applied [9]. This procedure is executed one more time to calculate the backup LP, once the links used in the primary LP are pruned from the topology (a necessary step to achieve link disjointness).

5.1.4 Performance Evaluation Metrics

A number of metrics can be used to evaluate the performance of survivable IA-RWA algorithms. Moreover, these metrics can be used as parameters during the design phase of such algorithms. The most relevant metrics are [118]:

Vulnerability Ratio or QoT-Vulnerability

The probability that, in the case of a link failure, a preplanned backup LP cannot be restored due to unacceptable QoT;

Cascading Failure Vulnerability

The probability that a given LP become unusable due to physical impairments induced by the activation of pre-planned backup LPs;

Failure Ratio

It is defined as the ratio between the number of connections that are not recovered due to unacceptable QoT to the number of primary LPs affected by a link failure. It is averaged over all single link failures;

Running Time

The time needed to compute the LP from the instance of the setup request arrival. Interesting values are the average and worst case scenario.

5.2 Proposed Algorithms

Three novel IA-RWA algorithms for WRPNs were developed, taking into account the needs of both future advanced applications and carriers operators. All of them are suitable to be integrated in PCE entities presented in GMPLS CPs. In fact, a single instance of a PCE can implement all three proposed algorithms, and LP setup requests over time can be fulfilled using different IA-RWA algorithms, at the discretion of the network administrator or the PCE itself.

The first algorithm, is a fast, online IA-RWA that assures the QoT necessary to satisfy the requested end-to-end bandwidth and BER. It performs on-the-fly multipath RWA calculation, and LP is selected by a multi-criteria rule set. Hence, this algorithm is referred as MCP-RWA (*Multipath CLA power-aware RWA*). As main novelties, it introduces the use of optical power and wavelength residual capacities in the combined RWA procedure, as well as the CLA (*Critical Link Avoidance*) feature. These novelties enhance traffic balance and network resource utilization. The other two algorithms are variants of MCP-RWA, that were enhanced to support critical services. In the IA-RWA context, critical services are those that must not be disrupted by QoT fluctuations and/or network failures. A WRPN, whose LPs are critical, must implement an IA-RWA algorithm that guarantees absolute levels of QoT during the setup phase, assuring with 100% probability that the LPs directly affected by the failure will be restored, and also that no pre-established LP will suffer from QoT penalties. At the best of the authors' knowledge, there is no IA-RWA specifically developed for critical services support. Usually, the main concern of survivable IA-RWA algorithms is to maximize the network throughput providing the best possible QoT, but without any guarantees that the LP requested QoT will be satisfied.

The primary objective of the two survivable variants of MCP-RWA is to offer either 1+1 or 1:1 protection, assuring the absolute levels of requested QoT both to the primary and secondary paths, even after a single link failure. As a second goal, these algorithms try to optimize the resource allocation as well as minimize the blocking probability of future requests without demanding a massive processing load. The first survivable IA-RWA, called MCP-D², uses a multipath RWA combined calculation based on the Dijkstra algorithm, while in the second one (MCP-S) the Suurballe algorithm [119] is used.

5.2.1 MCP-RWA

The MCP-RWA algorithm is able to satisfy LP setup requests with strict bandwidth and BER requirements, to assure LP QoT and, at the same time, to minimize the blocking probability of future requests by maintaining acceptable levels of optical power as well as adequate OSNR throughout the WRPN domain. Three parameters must be specified in LP setup requests: the source-destination pair,

the requested bandwidth (usually 2.5, 10 or 40 Gb/s) and the BER, numerically expressed in function of the Q factor. In case of a successful operation, the proposed IA-RWA engine returns not only the path and the wavelength tuple that will compose the new LP, but also the transmission optical power that must be used in the tunable laser at the source node to assure QoT (respecting the requested bandwidth and BER). In addition, the estimated BER for the new LP (which is equal or less than the required BER) is made available.

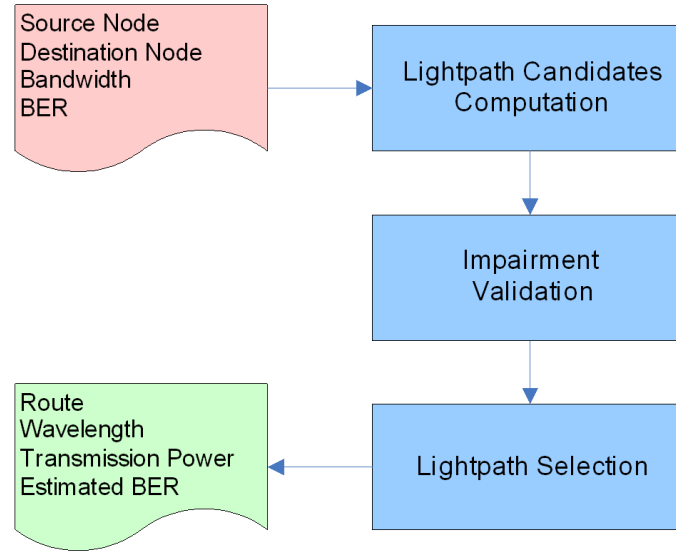


Figure 5.1: MCP-RWA macro-level flowchart.

At a macro-level, the proposed algorithm operation can be divided in three parts (see Figure 5.1): on-the-fly LP candidates computation (multipath IA-RWA with CLA, taking into account optical power and wavelength residual capacities); power-aware impairment validation (to discard candidates with inadequate QoT); and finally LP selection (to pickup the best candidate based on a multi-criteria rule set). All these steps are detailed in the following subsections.

Multipath RWA

The first part of the algorithm performs on-the-fly multipath RWA computation. Multipath techniques are commonly used to solve the routing subproblem offline, while just the wavelength assignment subproblem is solved online (usually with heuristics). Within the MCP-RWA context, multipath RWA means to calculate the best possible route for each usable wavelength. Considering the issues related to the actual use of all-optical converters (partial converters still have high costs, and full conversion is yet an immature technology), the wavelength continuity

5.2. Proposed Algorithms

79

constraint is applied.

In order to promote multipath RWA, the physical topology is described by a series of isolated wavelength planes, called WGs (*Wavelength Graphs*). Each WG describes the current topology view for a single wavelength. Given two adjacent nodes in a WG, for instance A and B, A is connected to B only if the wavelength for that particular WG is available in the fiber link from A to B. When a LP setup request arrives, the first step is to prune from the physical topology all WGs whose wavelengths at source node are unusable, saving processing time. Two conditions may render unusable a wavelength in a source node: the wavelength is being used on all fibers (just outgoing fibers for unidirectional LPs), or there is no available transponder for the requested bandwidth that can tune in that particular wavelength. After pruning, for each WG an instance of the Dijkstra's algorithm is executed. After this calculation, the minimum cost path of each WG is obtained. As the calculations of the paths in all WGs are completely independent, they are highly parallelizable, which leads to an optimization of the algorithm execution time.

Physical impairments are considered during the RWA process to weight links in WGs. Unlike the majority of RWA algorithms which use simple hop count (distance metric) to weight links [113], MCP-RWA uses an empirically defined impairment-aware link cost formulation. Due to the maximum power constraint, a request can be blocked even when there are continuously available wavelengths along a path for the requested endpoints. In some particular configurations, a single pre-established LP can use most of, or even all, the allowed optical power in a given fiber. Therefore, the residual power capacity, i.e., the optical power that still can be injected in a fiber, is an important metric to evaluate a link cost, together with the number of available wavelengths (or residual wavelength capacity). To find the best generic expression that calculates the link cost of a fiber in function of its wavelength and optical power residual capacities, simulations were carried out by taking into account a number of empirical formulas. Since the link cost must get higher and higher as its residual capacities decrease, the best expression is one where the link cost grows exponentially, i.e.:

$$f(P_{i,j}^{res}, \lambda_{i,j}^{res}) = round \left[10 \left(a^{\frac{1}{P_{i,j}^{res} \cdot \lambda_{i,j}^{res}}} \right) \right] \quad (5.1)$$

where $P_{i,j}^{res}$ and $\lambda_{i,j}^{res}$ are the power and wavelength residual capacities for the link (i, j) . The performance of Equation 5.1 are strictly related to the value of the exponential base, the a parameter. Thus, a set of simulations were performed in order to find which value of a would provide the top performance variant of the above expression (details in Paragraph 5.2.1).

Another strategy introduced by MCP-RWA to minimize the blocking probability is CLA. The "altruist" idea of avoiding using particular links to save them for future requests was introduced by the Asynchronous Criticality Avoidance (ACA) protocol [120]. Except for sharing this concept, CLA technique is completely different from ACA by any perspective. All links that are labeled as critical by the CLA are initially pruned from the physical topology. If, after the first attempt to find the minimum cost path in all WG that describe the topology, not even a single LP candidate is found, the process is repeated again, but this time considering all critical links that were not visible in the first pass.

The key aspect of CLA is the rule that defines the criticality of a link. For that purpose, the wavelength and power residual capacities were initially considered as candidate metrics. Different combinations of thresholds for these two metrics were used in simulations in order to find the best configuration for most WRPNS. The best results were found taking into account only the power residual capacity to define a link as critical, when 20% or less of the original capacity remains useable. The details are discussed in Paragraph 5.2.1.

Impairment Validation

When the multipath RWA phase of the algorithm ends, the impairment validation process of the LP candidates (described in the list of minimum cost paths from each WG) takes place. The impairment validation process of MCP-RWA relies upon an optical power-based impairment model that sets two conditions: the minimum and maximum power constraints. The minimum power constraint, which is best known as sensitivity level, assures that optical signals can be properly detected by all optical devices. The maximum power constraint limits the effects of fiber nonlinearities (which are power-dependent), because aggregate optical power on a link is restricted to a maximum value.

An analytical model to calculate the sensitivity level based on the ASE noise and the desired BER was introduced in [121]. This model is used by the MCP-RWA algorithm to guarantee that a LP can be established with the requested bandwidth and also with a maximum absolute BER value. The BER can be numerically evaluated in function of the Q factor [20]:

$$BER(Q) = \int_Q^\infty \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}} dx \quad (5.2)$$

where $Q = \frac{I_1 - I_0}{\sigma_0 + \sigma_1}$, I_1 and I_0 denote photocurrent sampled by the receiver during a 1 bit and a 0 bit, respectively, σ_0 and σ_1 the standard deviation of the corresponding noises, which are assumed to be Gaussian. The Q factor is

5.2. Proposed Algorithms

81

commonly used in the receiver performance specification, because it is related to OSNR necessary to achieve a certain BER. For example, for a BER of 10^{-12} (a common requirement for contemporary WDM systems), the Q factor is approximately 7. The sensitivity level at each PXC is determined by the following equation:

$$P_{sen} = 4Q^2 N_{sp} h f_c B_e \left(1 + \sqrt{1 + \frac{\frac{B_e}{B_o} - \frac{1}{2}}{4Q^2}} \right) \quad (5.3)$$

where N_{sp} is the spontaneous emission factor, h is the Planck's constant, f_c is the frequency of the optical carrier ($h f_c$ is the energy of the photon), B_o is the optical bandwidth (which is at most the spacing of the frequency grid in WDM systems) and B_e is the electrical bandwidth of the low-pass filter after the photodetector. The sensitivity level of a LP (i.e., the sum of sensitivity levels of all optical network elements traversed by a LP) can be obtained by using the equivalent pre-amplifier model [121]. This model allows the calculation of a spontaneous emission factor equivalent to all amplifiers along a LP, to be directly used in Equation 5.3. Its calculation is an iterative process which takes into account the whole amplifier cascade, starting from the pre-amplifier at the receiver (as shown in Figure 5.2):

$$N_{sp1}^{eq} = \frac{N_{sp1} (G_1 - 1) L_1 G_0 + N_{sp0} (G_0 - 1)}{G_1 L_1 G_0 - 1} \quad (5.4)$$

where N_{sp} , L and G are the spontaneous emission factor, the attenuation and gain of the involved amplifiers, respectively.

In the first iteration, the variables with an index equal to 0 are related to the pre-amplifier at the receiver, and the ones with an index equal to 1 are related to the amplifier immediately before to it. It is worth to note that the impairment validation used in this work considers the amplifiers as ideal devices with a stable gain. The influence of amplifiers with Automatic Gain Control is not assessed. Other than amplifiers, a WRPN node is composed by a number of components, such as taps, (de)multiplexers, switching matrix fabric, and so on. When an optical signal enters a node, it runs across such components and as a result there is a gain or loss of power. The power loss caused by a WRPN node can be derived with the following expression:

$$L_{sw} = 2 [\log_2 (D_i)] L_s + 4L_w \quad (5.5)$$

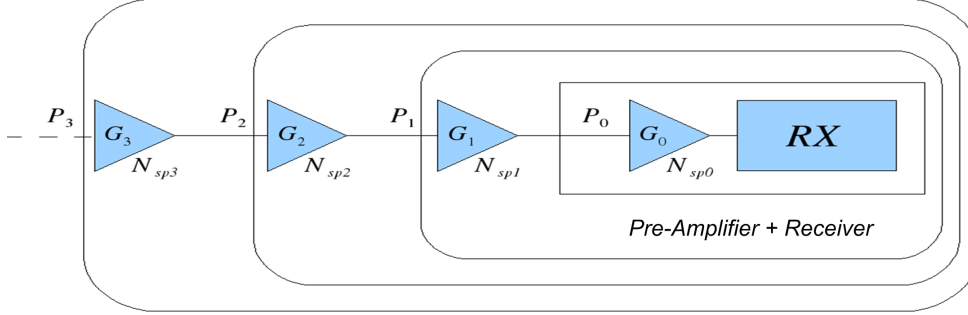


Figure 5.2: Cascading of amplifiers.

where D_i is the node degree (i.e., the number of links and stations attached to this node), L_s is the loss caused by the switching element insertion and L_w is the waveguide/fiber coupling loss.

The impairment validation process of RWA is divided in two steps: optical power estimation and pre-setup evaluation. In the first step, for each single LP candidate the minimum transmission power necessary to guarantee the required Q factor is calculated (i.e., the sensitivity level). Also, the fraction of the original transmission power for each link of the path is computed. The second step consists in verifying the feasibility of the LP, which means to check that all links that compose the LP can accommodate their share of optical power. The maximum power constraint must be respected. If even a single link fails to comply with this restriction, the LP is discarded.

LP Selection

The last phase of the algorithm is the final LP selection, that consists in selecting the best LP candidate among those whose QoT is already assured in the previous phase, respecting the bandwidth and Q factor specified in the setup request. At this point, all candidates (now described by path / wavelength / transmission power tuples) satisfy the setup request. Hence, the best candidate is such that, after its successful establishment, the WRPN is in a state where the blocking probability of future requests is minimized. To effectively choose the best candidate, a number of simple heuristics were considered. Through simulations, the relevance of these (isolated and combined) heuristics was analyzed. The best results were obtained using a multi-criteria rule set, evaluated in the following order:

1. lowest number of critical links;

5.2. Proposed Algorithms

83

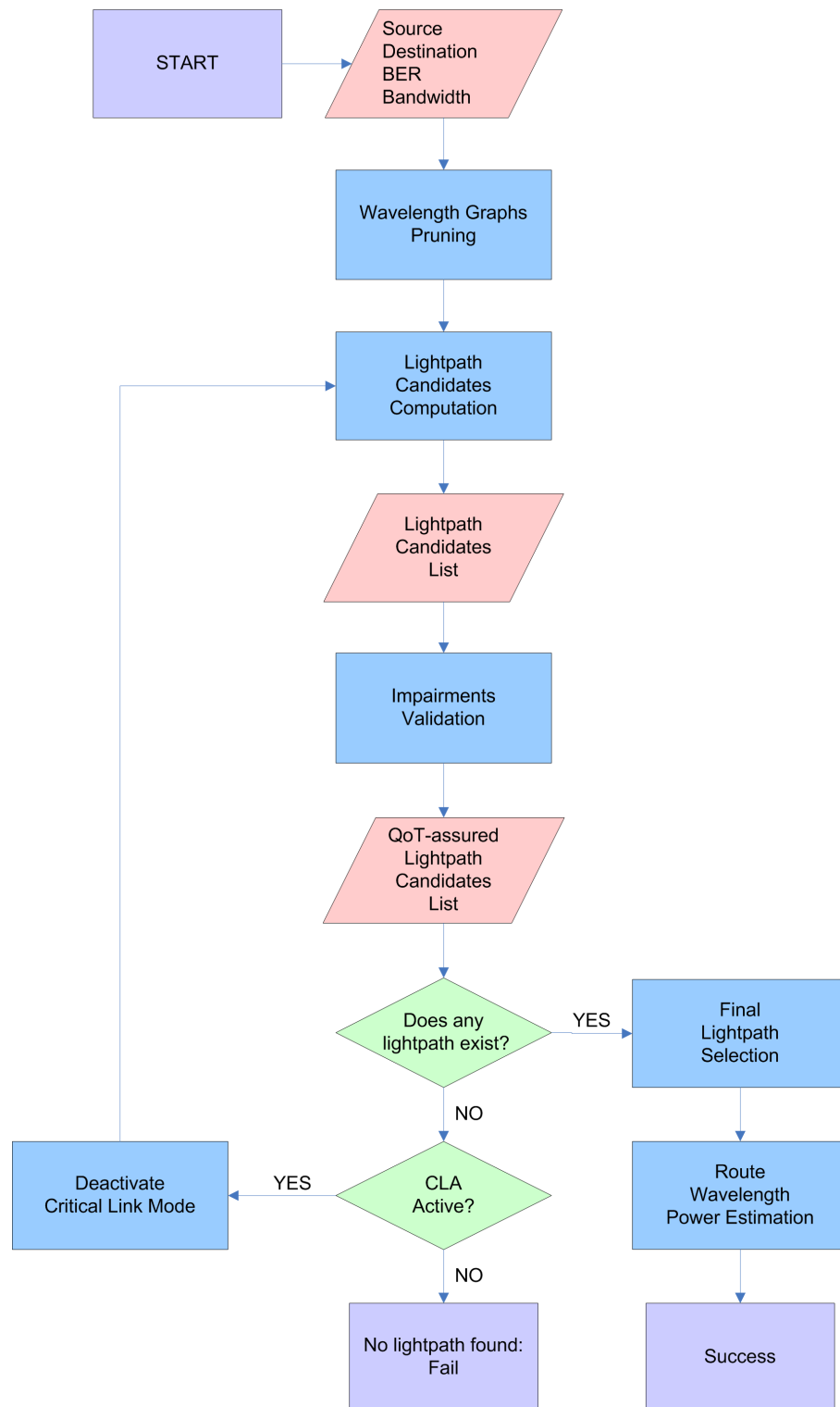


Figure 5.3: MCP-RWA detailed flowchart.

2. LP whose wavelength is the MU;
3. lowest transmission power;
4. lowest number of hops;
5. LP whose wavelength is the FF.

When more than one candidate matches a criterion, the matching ones are compared on the basis of the next criterion. If a criterion is matched by only one candidate, that LP is chosen to be established. The last criterion in the list guarantees that the selection process will always end up with precisely one LP, and is seldom matched. It is worth to mention that rule number 1 is ignored if all LP candidates were calculated when CLA was active, i.e. without using critical links.

The complete MCP-RWA algorithm is fully detailed in the flowchart presented in Figure 5.3.

MCP-RWA Validation

A simulated network scenario was built to design and refine the impairment-aware link cost function of the MCP-RWA algorithm. It was also used to evaluate the CLA optimization in function of diverse definitions of link criticality, based on different metrics combinations and thresholds. At last, the simulation environment was used to compare the performance of MCP-RWA with two other IA-RWA algorithms, using the mean blocking probability and the processing time as evaluation metrics. Simulations involving WRPNs are usually performed using classic, real-world topologies like the NSFNET [122] and the Italian High Speed Network [123], with 14 and 21 nodes respectively. In order to avoid polarization of results due to singularities of real-world and random topologies, it was chosen for the simulations an uniform 7 x 7 Manhattan topology. This simulated WRPN is therefore composed by 49 nodes, interconnected by pairs of unidirectional fibers. The chosen topology has more than the double of the number of nodes of classic topologies, as expected for future WRPNs. All links have a fixed length of 150 km, with inline amplifiers at each 50 km that have a constant gain of 11 dB. Optical transmitters have an operational power ranging from -20 to +15 dBm, and they are capable of tuning in 16 different wavelengths. For the simulations, the maximum optical power allowed per channel was set to 9 dBm, while the maximum total power per link was set to 12 dBm. Connection requests are generated with randomly chosen source-destination pairs, with a bandwidth of 10 Gb/s and a Q factor equal to 7 (BER 10^{-12}).

In the next paragraphs, design decisions and the MCP-RWA evaluation are drawn. For the sake of clarity, graphs only show the most relevant curves obtained from simulations.

5.2. Proposed Algorithms

85

Impairment-aware Link Cost Function The effectiveness of Equation 5.1 as the impairment-aware link cost function for MCP-RWA was evaluated taking as reference function the simple hop count link cost. A series of simulations was performed using as cost functions both the reference one and Equation 5.1, considering different values for the a parameter. It was found that the proposed link cost function based on residual optical power and wavelength capacity performs better than simple hop count, when $1 < a < 2$. The maximum efficiency was obtained when $a = 1.1$. Figure 5.4 shows the blocking probability obtained for the most significant values of a .

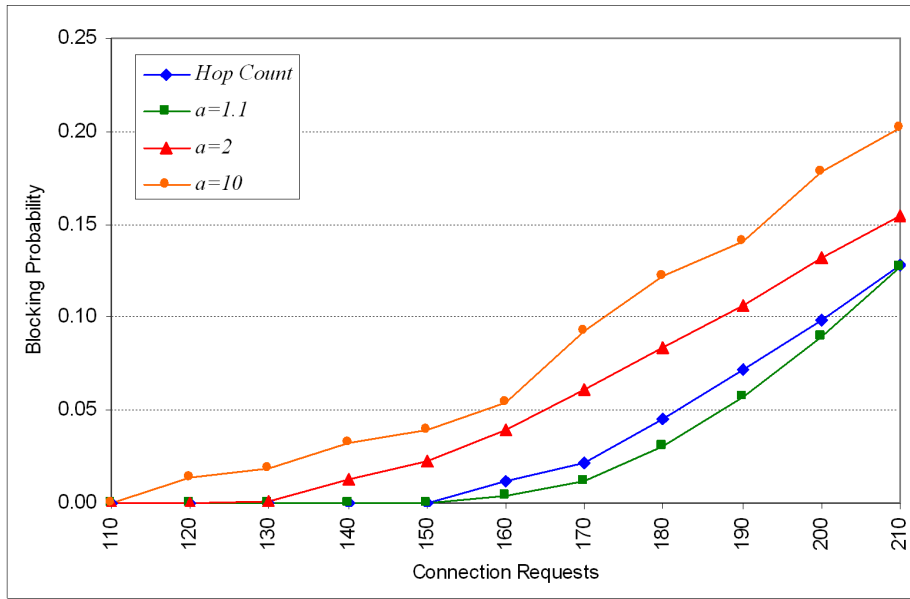


Figure 5.4: Cost function comparison.

Criticality Thresholds In order to find the most appropriate rule to define a link as critical during the CLA process, several simulations were carried out using power and wavelength residual capacities as metrics, and also different thresholds as lower bounds for these metrics. Figure 5.5 shows how MCP-RWA performs when CLA is deactivated and when CLA is operating using the following conditions (criticality thresholds) to set a link as critical:

- wavelength residual capacity is equal or less than 20%;
- optical power residual capacity is equal or less than 20%;
- either power or wavelength capacities are equal or less than 20%.

The best performance is attained when CLA is active and only the power residual capacity, at a rate of 20% or less, is considered to define a resource as critical.

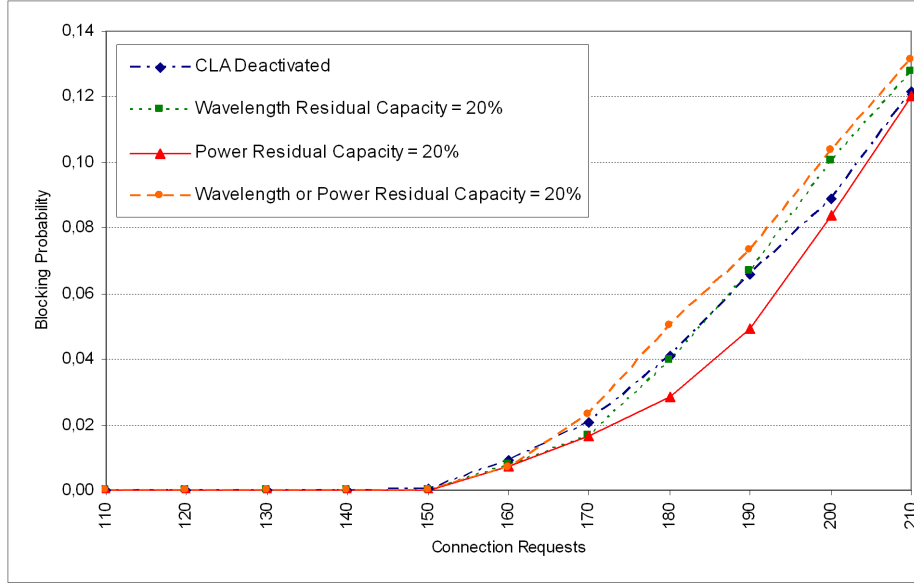


Figure 5.5: Criticality threshold comparison.

It is worth to be noted that an inappropriate configuration of CLA parameters can led to poor algorithm performance.

Overall MCP-RWA evaluation To validate the overall performance of MCP-RWA, it was compared with two other IA-RWA algorithms in the above described scenario. For fairness of comparison, all the IA-RWA approaches were implemented using the same impairment validation process of MCP-RWA, as described in Subsubsection 5.2.1. Furthermore, all IA-RWA algorithms (MCP-RWA included) require the same physical topology information knowledge to operate properly. In the first approach, the routing, wavelength assignment and impairment validation processes are completely decoupled from each other. The well known Yen’s FAR algorithm [124] is used to offline calculate 3 shortest paths for all source-destination pairs. When a connection request arrives, the MU heuristic is employed to assign a wavelength to the candidate paths. The connection request can not be satisfied if there is no continuous wavelength available in any previously calculated path, or if the selected LP can not offer the required level of QoT. This strategy (FAR+MU) is the same as the one proposed in [63], except for the wavelength assignment heuristics (MU instead FF). The second approach (MU+WG) also employs the MU heuristic to perform wavelength

5.2. Proposed Algorithms

87

assignment, and uses WGs to online calculate LPs as MCP-RWA does, but with a few limitations. The simple hop count link cost function is used to weight links, and CLA is not present. When a connection request arrives, the shortest path is calculated in the WG whose wavelength is the first one found by the MU heuristic. If it is not possible to find a path, the next wavelength plane is used, always as defined by the MU heuristic. Setup fails if no path can be calculated in any WG, or if any calculated LP can not offer QoT based on the required BER and bandwidth. Simulation results are shown in Figure 5.6, Figure 5.7 and Figure 5.8. The mean blocking probability (Figure 5.6) and the width of the 95% confidence intervals (Figure 5.7) are presented in separate figures for the sake of clarity. As expected, FAR+MU presents the worst performance regarding the blocking probability, caused by the high wavelength fragmentation (an undesirable consequence of path computation techniques that do not take into account wavelength availability). The other two approaches that rely on WGs by far outperform FAR+MU.

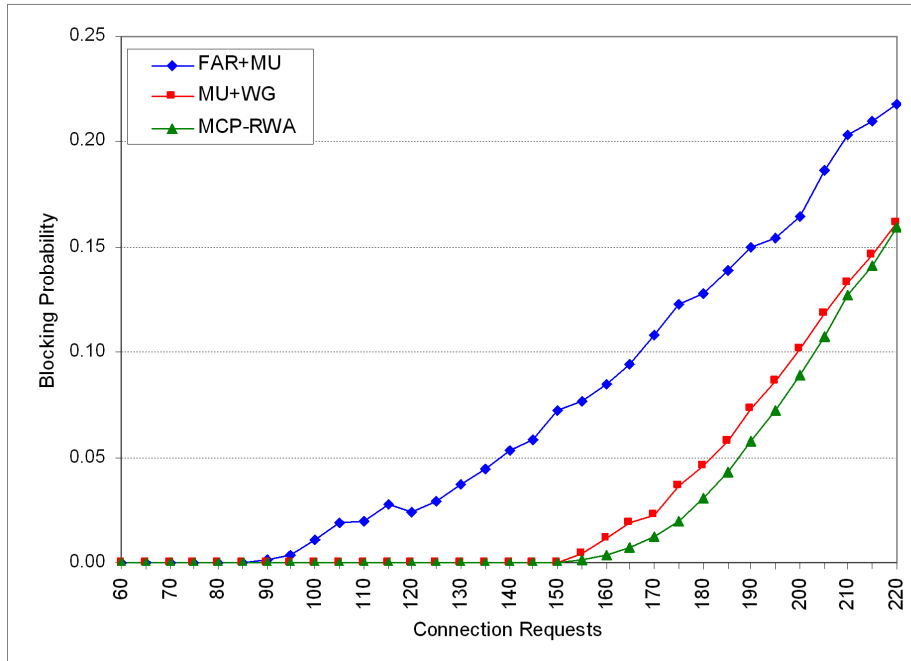


Figure 5.6: MCP-RWA mean blocking probability.

Figure 5.6 also shows that MCP-RWA presents better performance than WG+MU due to the multipath RWA, the impairment-aware link cost function, the CLA technique and the multicriteria rule set. Moreover, as reported in Figure 5.7, the introduction of the impairment-aware link cost function in MCP-RWA narrows and makes less variable the confidence intervals of the blocking probability, which is

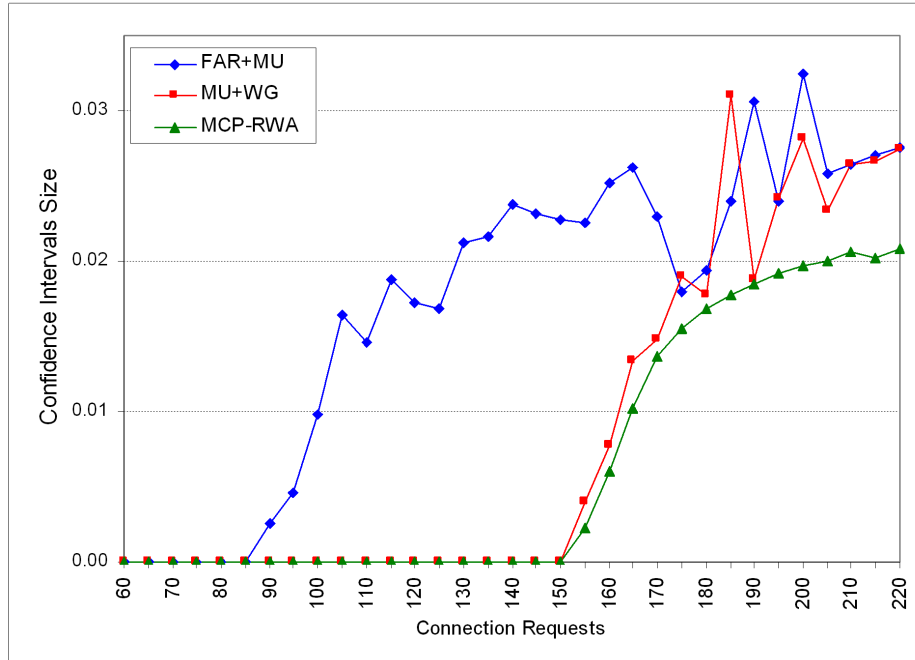


Figure 5.7: Confidence intervals for MCP-RWA mean blocking probability.

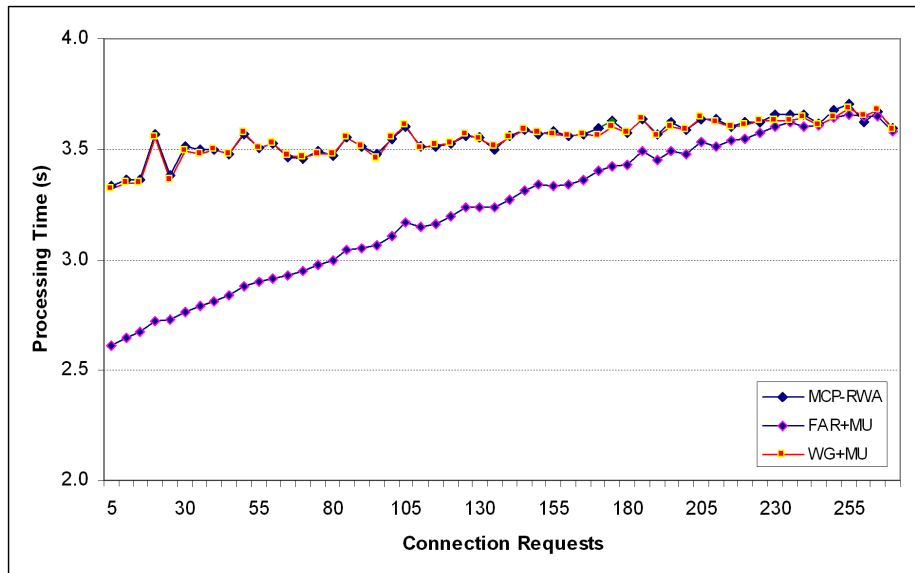


Figure 5.8: MCP-RWA processing time.

5.2. Proposed Algorithms

89

a desired feature.

Figure 5.8 depicts the average processing time needed by the algorithms in order to comply with the set of requests. The performance of FAR+MU degrades proportionally to number of connection requests due to reduction of the available resources for LP setup, and eventually becomes the worst. The processing time used by MCP-RWA is slightly superior of the WG+MU one. Thus, the benefits of MCP-RWA over WG+MU have almost no impact in the processing time needed to satisfy setup requests.

5.2.2 MCP-D²

As stated before, LPs used for critical services must have assured QoT and survivability, even in case of network failures. These requirements imply that the QoT Vulnerability, the Cascading Failure Vulnerability and consequently the Failure Ratio must be equal to zero. Therefore, as restoration techniques can not fulfill these requirements, MCP-D² (as well as MCP-S, described in the next Subsection) strictly provides dedicated 1+1/1:1 protection only. Moreover, as single fiber cuts are the most common type of failures, and WRPNs for critical services are usually designed with redundant hardware, the proposed survivable IA-RWA algorithms aim at guaranteeing QoT and survivability only in case of a single link failure. This way the complexity of the proposed algorithms is much lower, and the LPs are not subjected to great setup delays.

Roughly speaking, MCP-D² calculates a primary and a secondary LP by applying twice the MCP-RWA algorithm, first to the original topology and then to the residual network graph after all links belonging to the primary LP have been pruned. In case one of the LPs blocks, the request for a protected LP cannot be satisfied. The MCP-D² detailed flowchart is shown in Figure 5.9.

5.2.3 MCP-S

In this case, the RWA procedure uses the Suurballe algorithm to calculate a pair of disjoint paths on each available WG. Theoretically, using two disjoint LPs of the same wavelength as primary and secondary LPs decreases the blocking probability of future requests by avoiding the “trap topology” problem [119], and by reducing the wavelength fragmentation phenomenon [20]. Moreover, if the MCP-S RWA procedure initially fails to find a pair of disjoint paths for a given WG, then it tries to calculate at least one path using the Dijkstra algorithm.

When the RWA procedure ends, all LP candidates are subjected to impairments validation. The LPs, whose QoT does not meet the input requirements, are removed from the candidates list. At this point, the final selection procedure starts.

1. For each WG that contains at least one LP, a group is created. LPs from the

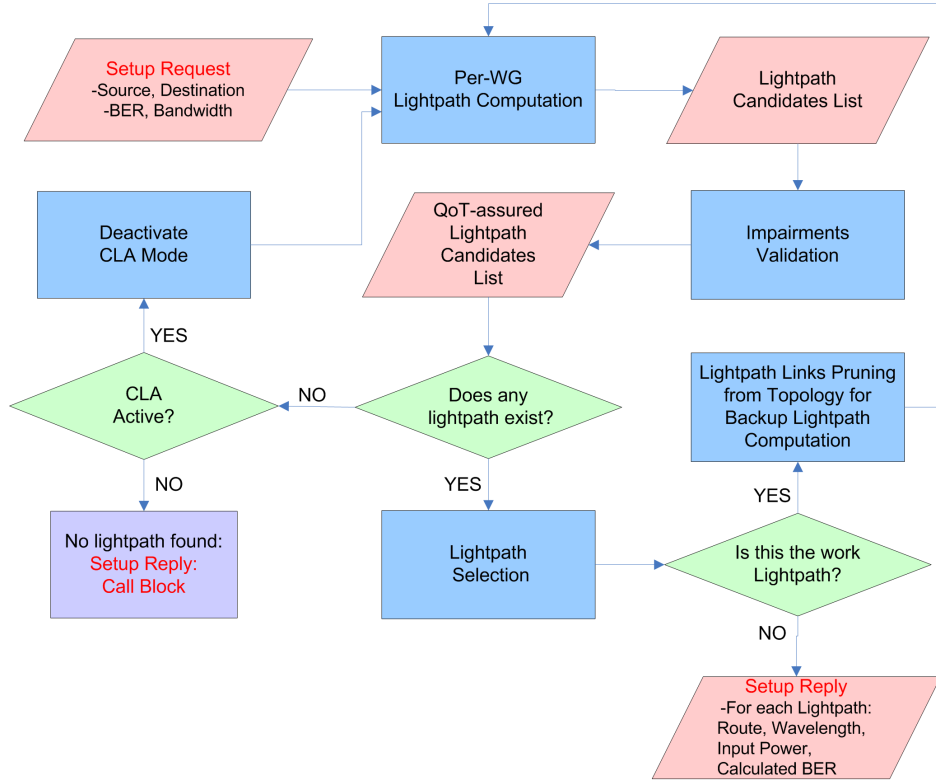


Figure 5.9: MCP-D² detailed flowchart.

same WG are assigned to a specific group. Within each group, either the LP with the shortest path obtained by means of the Suurballe algorithm, or the LP found with the Dijkstra algorithm, is called group-head.

2. LPs from the other WG that are link disjoint with the group-head are added to each group.
3. The best group is selected according to the following ordered criteria (the last criterion guarantees that only one group is chosen):
 - (a) group-head with the lowest number of critical links (only when CLA is deactivated);
 - (b) group with an LP of the same wavelength as its group-head, only if its number of hops is no more than 25% higher than the group-head number of hops;
 - (c) group-head with the lowest number of hops;
 - (d) group-head with the lowest inject power;

5.2. Proposed Algorithms

91

- (e) group-head whose wavelength is given by the FF with Ordering (FFwO) [13].
4. The group-head of the chosen group is elected as primary LP. The secondary LP is chosen among the remaining LPs of the group based on the following criteria:
 - (a) LP with the lowest number of critical links (only when CLA is deactivated);
 - (b) LP with the lowest number of hops;
 - (c) LP with the lowest injection power;
 - (d) LP whose wavelength is given by FFwO.

The MCP-S flowchart is shown in Figure 5.10.

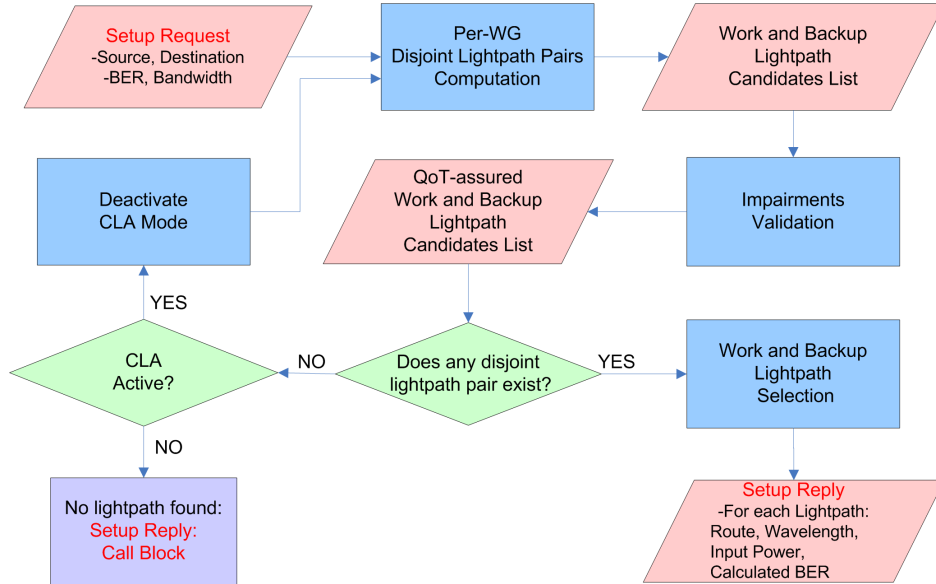


Figure 5.10: MCP-S detailed flowchart.

MCP-D² and MCP-S Validation

The performance of the proposed algorithms have been evaluated in comparison with (D+FFwO)², which basically uses the following procedure:

1. the primary LP is computed by using the Dijkstra algorithm;
2. a wavelength is assigned to the primary LP using the FFwO heuristic and then its QoT is evaluated;

3. the primary LP links are pruned from the topology;
4. the backup LP is computed and impairment validated repeating the same procedure used in the initial phase.

For the sake of fairness in the comparisons, $(D+FFwO)^2$ was implemented using the same network impairment model used by MCP-D2 and MCP-S.

Simulations were carried out using four different network topologies, each one with specific characteristics:

- a uniform 7 x 7 Manhattan (or grid) topology, chosen to avoid polarization of the results due to singularities of random and real-world topologies. For this 49-node topology, all 82 links have a fixed length of 150 km;
- the classic 14-node, 21-link NSFNET with a size scaled down to 1:10 (length of the links varies from 60 to 280 km);
- the high-speed all-optical Italian Network, a 21-node, 33-link mesh topology. Due to its north-south orientation, links failures at the center of the topology can severely disrupt communications. All links have the same length as the original network, ranging from 55 to 460 km;
- a large 24-node, 43-link heavily meshed topology from an American telecommunication carrier [125]. Its links have a variable length from 50 to 250 km. WRPNS build to support future Internet applications are likely to have topologies of this type.

For all topologies, inline amplifiers with a constant gain of 11 dB are placed in links at each 50 km on average. Transponders have an operational power ranging from -20 to +15 dBm, and are capable of tuning in 16 different wavelengths. For the simulations, the maximum optical power allowed per channel was set to 9 dBm, while the maximum total power per link was set to 12 dBm. Connection requests were generated with randomly chosen source-destination pairs, with a bandwidth of 10 Gb/s and a Q factor equal to 7 (BER 10^{-12}). The chosen performance metrics are the blocking probability and the processing time in function of the number of requests. For each value of the connection requests, simulations have been repeated 10 times and average values are reported in Figures 5.11 through 5.18.

As far as the blocking probability is concerned, MCP-D² outperforms both $(D+FFwO)^2$ and MCP-S except for the American Network topology when the connection requests number is less than 90. MCP-S assures a lower blocking probability than $(D+FFwO)^2$ for the NSFNET topology as well as for the Italian Network and American Network topologies when the connection requests number is not too large. On the whole, when the network is unloaded, i.e., when the

5.2. Proposed Algorithms

93

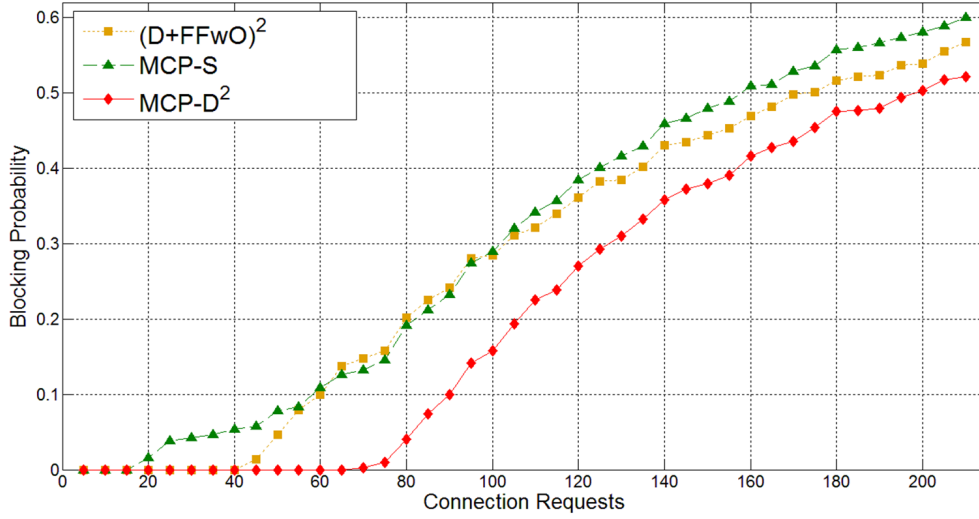


Figure 5.11: Survivable IA-RWA grid topology blocking probability.

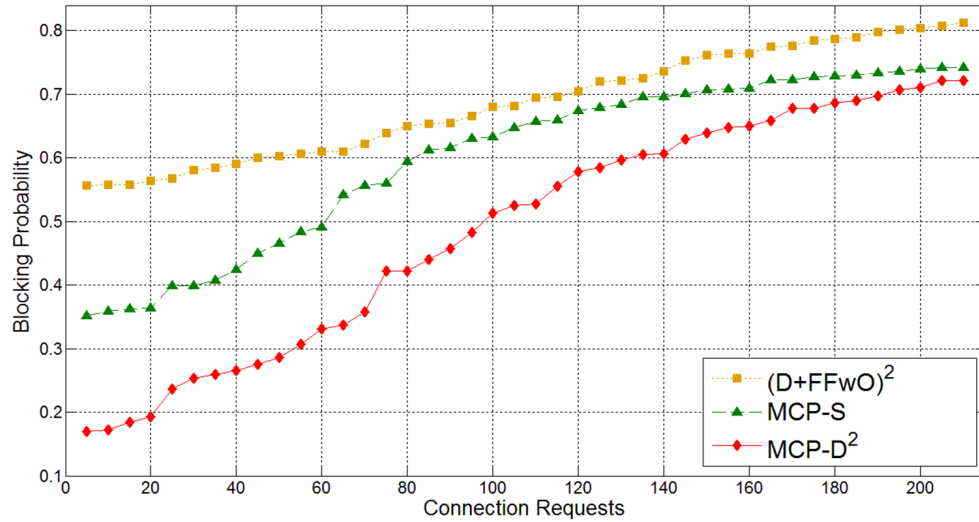


Figure 5.12: Survivable IA-RWA NSFNET topology blocking probability.

wavelengths availability on links is large and the total optical power injected into the links is not near the maximum allowed value, MCP-S and MCP-D² guarantee the best performance. Since a grid network has a regular, very meshed topology with several paths available for every source-destination node pairs, the advanced features of MCP-S are not useful to reduce the blocking probability as compared with (D+FFwO)². However, considering real-world, asymmetric topologies, such as the NSFNET, the Italian Network and American Network ones, where nodes

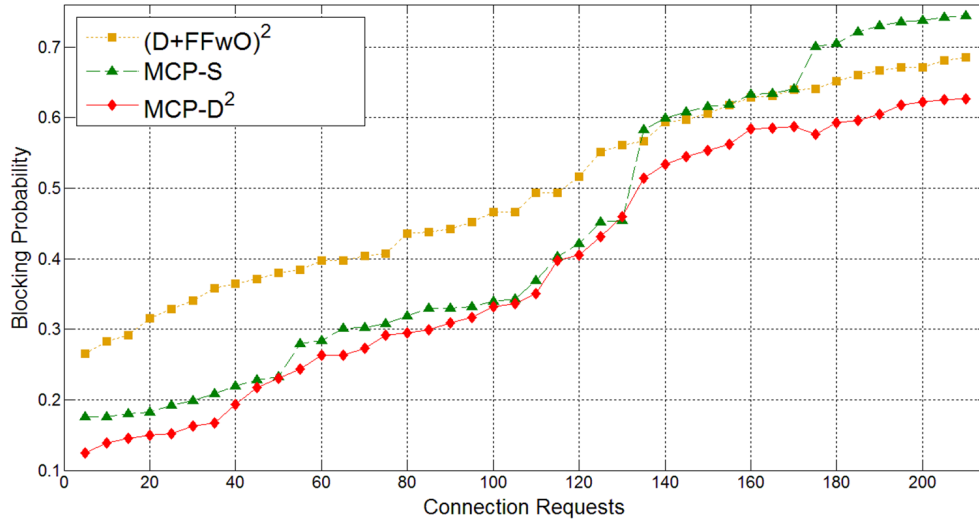


Figure 5.13: Survivable IA-RWA Italian topology blocking probability.

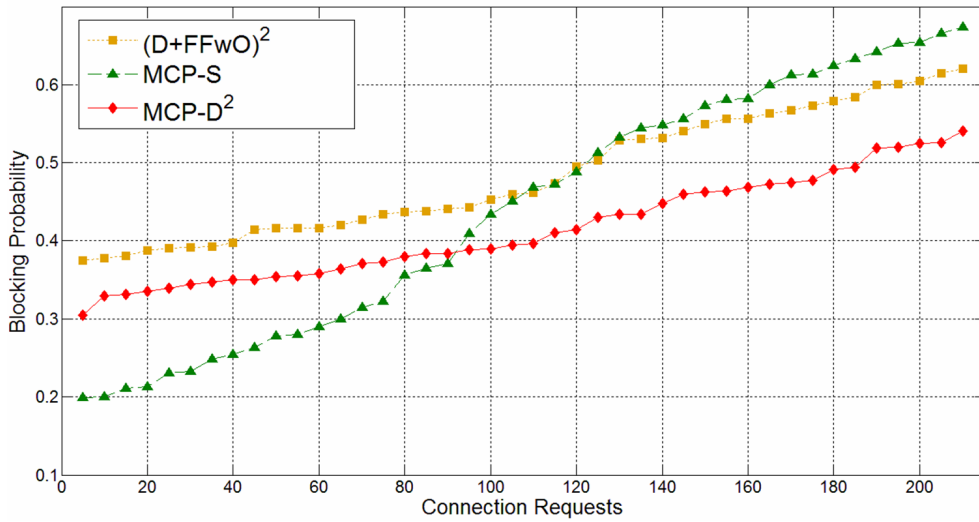


Figure 5.14: Survivable IA-RWA American topology blocking probability.

have a highly variable number of links, the advantage of using MCP-D² and MCP-S over (D+FFwO)² is noticeable.

Regarding the processing time, some relevant conclusions can be drawn. When the connection requests number is low, (D+FFwO)² has the lowest processing time regardless of the network topology. Moreover, as expected, with (D+FFwO)² the processing time required to complete the calculations increases when the connection requests number and, as a consequence, the already established

5.2. Proposed Algorithms

95

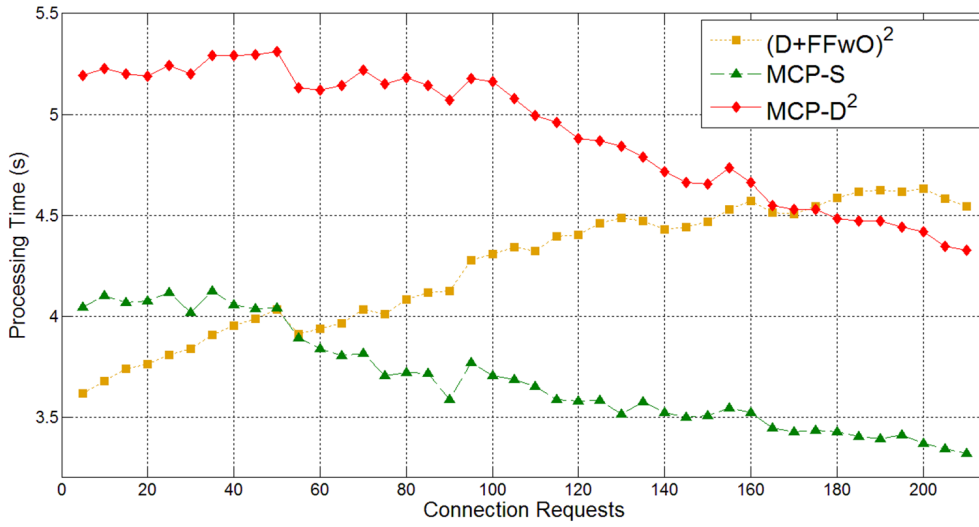


Figure 5.15: Survivable IA-RWA grid topology processing time.

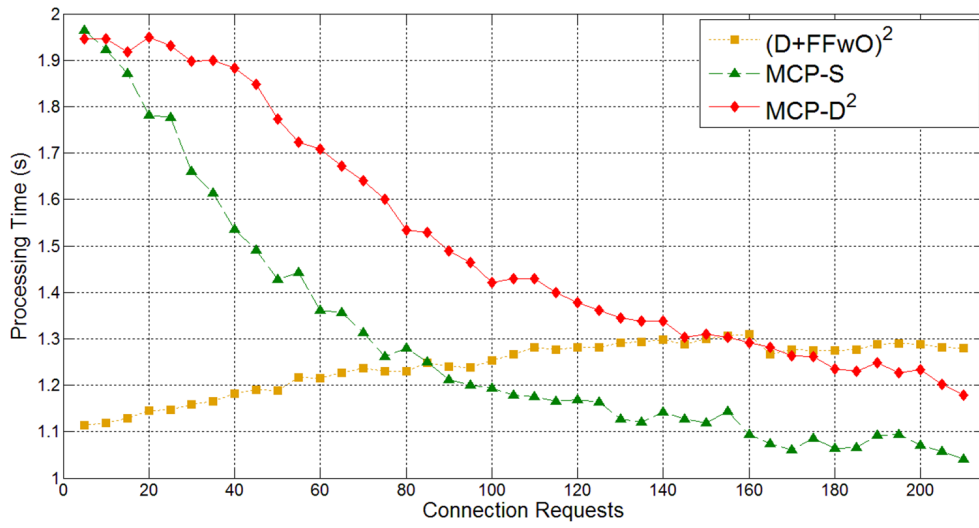


Figure 5.16: Survivable IA-RWA NSFNET topology processing time.

protected LPs number, increases. Instead, the processing time for the new proposed algorithms has a completely different behaviour and decreases when the number of connection requests increases. This is due to the IA-RWA procedure of MCP-D² and MCP-S, that subtly discards unusable WG. In general, MCP-D² requires the highest processing time (except in case of grid and NSFNET topologies for the highest number of connection requests), whereas MCP-S assures the best performance especially when a high number of LPs has to be

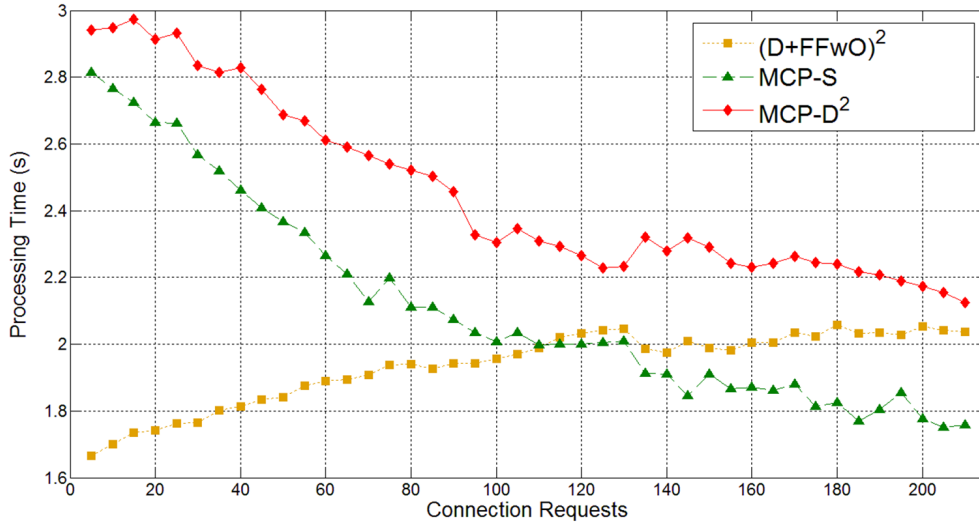


Figure 5.17: Survivable IA-RWA Italian topology processing time.

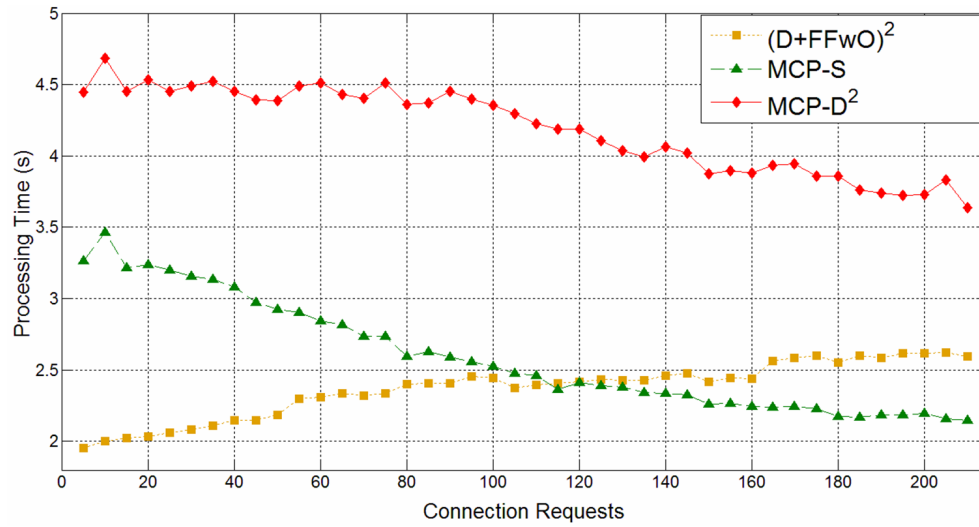


Figure 5.18: Survivable IA-RWA American topology processing time.

setup.

Finally, it is worth highlighting that the processing time necessary to compute the path is just a fraction of the overall LP setup time, which also includes the signaling time and the configuration time of all the network elements along the path. Thus, MCP-D² seems to be the best choice because it assures the lowest blocking probability and, at the same time, the higher processing time is mitigated when the overall setup time is considered. Moreover, when the WRPN has a

5.2. Proposed Algorithms

97

really dense wavelength pool (i.e., the spacing between wavelengths is short), crosstalking has a strong impact on the final QoT, thus MCP-S is a more attractive choice.

Chapter 6

Conclusion

The Internet “just works”. Victim of its own success, the Internet of today is limited by its original architecture, unable to incorporate at global scale the recent advances in telecommunications, computing and human-machine interaction. These paradigm shifting technologies will require from the network a complete new set of functionalities in order to be deployed. NGNs are being envisioned to precisely offer these functionalities. NGN visionaries concentrate their efforts to identify the requirements to support highly advanced and profitable applications, and the means to provide them. However, in order to meet the needs of future applications and unleash a novel online experience, NGNs must rely on intelligent telecommunication infrastructures, that are able to quickly deploy services and perform data delivery with guaranteed quality levels, in a efficient manner. This thesis has presented three proposals towards enabling the future Internet, focusing at the infrastructure level.

The first proposal regards a preliminary architecture to provide NGNs with a flexible, on demand infrastructure service over an alliance of transport networks. The alliance of federated domains is virtualized as a single network infrastructure, and client NGNs can manage the alliance concurrently, as it was not shared by all clients, without having to deal with interdomain intricacies. This is achieved by adding an upper control layer that uses modified instances of GMPLS protocols. This layer is also responsible for offering services that each alliance member usually can not perform by itself, like MP2P links and on-the-fly cryptography. By using the TNVE, NGNs can focus on user services, while relying on a robust, scalable infrastructure. One key aspect that differentiates the TNVE is its business model, that directly incorporates the monetary cost in the interdomain relationship dynamics, inspired on the success of the BGP4 and its policing rules. The second proposal introduces a novel framework exclusively based on open source software, that uses commodity PC hardware to create DS- and MPLS-enabled FSRs. Diversely from other SR implementations that can be

summarized as collections of tools, the proposed DS-MPLS framework tightly integrates the open source software components to allow a highly customizable LSP provisioning. By using the featured CP entities, uni- and bidirectional circuits can be created effortlessly. Moreover, live distributions were developed, allowing not only the addition or substitution of a real DS-LSR, but also an entire fully-functional DS-MPLS virtual network within minutes. These live distributions are of immense value to act as a QoS platform provider supporting the development of NGNs, and also to support advanced networking teaching activities for graduate courses. The open framework is in constant development. In the next development cycles, the OSPF-TE and RSVP-TE daemons are expected to be completed, and some initial SSMs services are being considered. The third proposal introduces three new online IA-RWA algorithms specifically tailored for future Internet services provisioning in advanced WRPNS. The algorithms were designed to assure absolute QoT and 100% survivability in case of single failures, with the aim to minimize the resource utilization and the blocking probability of future requests, and without incurring in longer setup delays. These goals were achieved by combining a high parallelizable multipath IA-RWA procedure (based on Dijkstra and Suurballe algorithms) with simple but effective heuristics. The performance of the newly proposed algorithms were evaluated in comparison with (D+FFwO)² that, at the best of the author's knowledge, is the sole algorithm that is capable of assuring absolute QoT and survivability. Simulations with different topologies show that the introduced algorithms achieve better blocking probability, and under certain conditions require even less processing time.

References

- [1] Q. Vohra and E. Chen, “BGP Support for Four-octet AS Number Space,” RFC 4893 (Proposed Standard), Internet Engineering Task Force, May 2007. [Online]. Available: <http://www.ietf.org/rfc/rfc4893.txt> 1
- [2] A. Nakao, L. Peterson, and A. Bavier, “A routing underlay for overlay networks,” in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03. New York, NY, USA: ACM, 2003, pp. 11–18. [Online]. Available: <http://doi.acm.org/10.1145/863955.863958> 2
- [3] A. Meddeb, “Internet QoS: Pieces of the puzzle,” *IEEE Communications Magazine*, vol. 48, no. 1, pp. 86–94, Jan. 2010. 2, 3, 34
- [4] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the internet impasse through virtualization,” *Computer*, vol. 38, pp. 34–41, Apr. 2005. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1058219.1058273> 2
- [5] “Available Pool of Unallocated IPv4 Internet Addresses Now Completely Emptied,” News Release, Feb. 2011, FOR IMMEDIATE RELEASE. [Online]. Available: <http://www.icann.org/en/news/releases/release-03feb11-en.pdf> 2
- [6] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification,” RFC 2460 (Draft Standard), Internet Engineering Task Force, Dec. 1998, updated by RFCs 5095, 5722, 5871. [Online]. Available: <http://www.ietf.org/rfc/rfc2460.txt> 2
- [7] J. Schönwälder, M. Fouquet, G. Rodosek, and I. Hochstatter, “Future Internet = content + services + management,” *IEEE Communications Magazine*, vol. 47, no. 7, pp. 27–33, Jul. 2009. 3
- [8] T. Wolf, “In-network services for customization in next-generation networks,” *IEEE Network*, vol. 24, no. 4, pp. 6–12, Jul./Aug. 2010. 3, 50

- [9] "Clean Slate Design for the Internet," Feb. 2011. [Online]. Available: <http://cleanslate.stanford.edu> 3
- [10] P. Castoldi, F. Baroncelli, B. Martini, V. Martini, and L. Valcarenghi, "Service plane: Capabilities and challenges," in *FEDERICA-Phosphorus Tutorial and Workshop*, May 2008. [Online]. Available: <http://www.ist-phosphorus.org/files/tnc2008workshop/pcastoldi-tutorial-tnc2008.pdf> 3
- [11] N. Chowdhury and R. Boutaba, "Network virtualization: state of the art and research challenges," *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20–26, Jul. 2009. 3, 4, 27, 28, 32
- [12] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," RFC 3945 (Proposed Standard), Internet Engineering Task Force, Oct. 2004, updated by RFC 6002. [Online]. Available: <http://www.ietf.org/rfc/rfc3945.txt> 4, 22
- [13] K.-i. Sato and H. Hasegawa, "Optical networking technologies that will create future bandwidth-abundant networks [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 1, no. 2, pp. A81–A93, Jul. 2009. 4
- [14] "Specifications of signaling system No. 7," ITU-T Recommendation Q.700, Mar. 1993. [Online]. Available: <http://www.itu.int/rec/T-REC-Q.700/en> 4
- [15] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271 (Draft Standard), Internet Engineering Task Force, Jan. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4271.txt> 4
- [16] R. Douville, J.-L. Le Roux, J.-L. Rougier, and S. Secci, "A service plane over the PCE architecture for automatic multidomain connection-oriented services," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 94–102, Jun. 2008. 4, 39
- [17] J. Wu, Y. Zhang, Z. M. Mao, and K. G. Shin, "Internet routing resilience to failures: analysis and implications," in *Proceedings of the 2007 ACM CoNEXT conference*, ser. CoNEXT '07. New York, NY, USA: ACM, 2007, pp. 25:1–25:12. [Online]. Available: <http://doi.acm.org/10.1145/1364654.1364687> 4
- [18] A. Farrel, J.-P. Vasseur, and J. Ash, "A Path Computation Element (PCE)-Based Architecture," RFC 4655 (Informational), Internet Engineering Task Force, Aug. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4655.txt> 4, 26

REFERENCES

103

- [19] J.-P. Vasseur, P. Demeester, M. Pickavet, and K. Johnson, *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Morgan Kaufmann, Aug. 2004. 8
- [20] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical Networks: A Practical Perspective*, 3rd ed. Morgan Kaufmann, Jul. 2009. 8, 80, 89
- [21] J. Strand and A. Chiu, “Impairments and Other Constraints on Optical Layer Routing,” RFC 4054 (Informational), Internet Engineering Task Force, May 2005. [Online]. Available: <http://www.ietf.org/rfc/rfc4054.txt> 9
- [22] A. Vannucci and A. Bononi, “A change of perspective on single- and double-stage optical PMD compensation,” *IEEE/OSA Journal of Lightwave Technology*, vol. 26, no. 14, pp. 2087–2097, Jul. 2008. 9
- [23] A. Erdogan, A. Demir, and T. Oktem, “Automatic PMD compensation by unsupervised polarization diversity combining coherent receivers,” *IEEE/OSA Journal of Lightwave Technology*, vol. 26, no. 13, pp. 1823–1834, Jul. 2008. 9
- [24] R. Martínez, C. Pinart, F. Cugini, N. Andriolli, L. Valcarenghi, P. Castoldi, L. Wosinska, J. Comellas, and G. Junyent, “Challenges and requirements for introducing impairment-awareness into the management and control planes of ASON/GMPLS WDM networks,” *IEEE Communications Magazine*, vol. 44, no. 12, pp. 76–85, Dec. 2006. 9, 73
- [25] G. Agrawal, *Nonlinear Fiber Optics*, 4th ed. Academic Press, Oct. 2006. 9
- [26] G. Bernstein, B. Rajagopalan, and D. Saha, *Optical Network Control: Architecture, Protocols, and Standards*. Addison-Wesley, Jul. 2003. 10
- [27] R. Muñoz, R. Martínez, and R. Casellas, “Challenges for GMPLS lightpath provisioning in transparent optical networks: Wavelength constraints in routing and signaling,” *IEEE Communications Magazine*, vol. 47, no. 8, pp. 26–34, Aug. 2009. 10
- [28] J. Strand, A. Chiu, and R. Tkach, “Issues for routing in the optical layer,” *IEEE Communications Magazine*, vol. 39, no. 2, pp. 81–87, Feb. 2001. 11
- [29] H. T. Mouftah and Pin-Han Ho, *Optical Networks: Architecture and Survivability*. Springer, Dec. 2002. 13
- [30] M. Ilyas and H. T. Mouftah, *The Handbook of Optical Communication Networks*. CRC Press, Apr. 2003. 13

- [31] C. Lefelhocz, B. Lyles, S. Shenker, and L. Zhang, "Congestion control for best-effort service: why we need a new paradigm," *IEEE Network*, vol. 10, no. 1, pp. 10–19, Jan./Feb. 1996. 14
- [32] J. Postel, "Internet Protocol," RFC 791 (Standard), Internet Engineering Task Force, Sep. 1981, updated by RFC 1349. [Online]. Available: <http://www.ietf.org/rfc/rfc791.txt> 14
- [33] P. Almquist, "Type of Service in the Internet Protocol Suite," RFC 1349 (Proposed Standard), Internet Engineering Task Force, Jul. 1992, obsoleted by RFC 2474. [Online]. Available: <http://www.ietf.org/rfc/rfc1349.txt> 14, 15
- [34] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633 (Informational), Internet Engineering Task Force, Jun. 1994. [Online]. Available: <http://www.ietf.org/rfc/rfc1633.txt> 15
- [35] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification," RFC 2205 (Proposed Standard), Internet Engineering Task Force, Sep. 1997, updated by RFCs 2750, 3936, 4495, 5946. [Online]. Available: <http://www.ietf.org/rfc/rfc2205.txt> 16
- [36] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Service," RFC 2475 (Informational), Internet Engineering Task Force, Dec. 1998, updated by RFC 3260. [Online]. Available: <http://www.ietf.org/rfc/rfc2475.txt> 16
- [37] D. Grossman, "New Terminology and Clarifications for Diffserv," RFC 3260 (Informational), Internet Engineering Task Force, Apr. 2002. [Online]. Available: <http://www.ietf.org/rfc/rfc3260.txt> 16, 17
- [38] A. Leon-Garcia and I. Widjaja, *Communication Networks*, 2nd ed. McGraw-Hill, Jul. 2003. 16
- [39] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474 (Proposed Standard), Internet Engineering Task Force, Dec. 1998, updated by RFCs 3168, 3260. [Online]. Available: <http://www.ietf.org/rfc/rfc2474.txt> 17
- [40] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," RFC 3168 (Proposed Standard), Internet Engineering Task Force, Sep. 2001, updated by RFCs 4301, 6040. [Online]. Available: <http://www.ietf.org/rfc/rfc3168.txt> 17

REFERENCES

105

- [41] J. Babiarz, K. Chan, and F. Baker, "Configuration Guidelines for DiffServ Service Classes," RFC 4594 (Informational), Internet Engineering Task Force, Aug. 2006, updated by RFC 5865. [Online]. Available: <http://www.ietf.org/rfc/rfc4594.txt> 17
- [42] V. Jacobson, K. Nichols, and K. Poduri, "An Expedited Forwarding PHB," RFC 2598 (Proposed Standard), Internet Engineering Task Force, Jun. 1999, obsoleted by RFC 3246. [Online]. Available: <http://www.ietf.org/rfc/rfc2598.txt> 17
- [43] B. Davie, A. Charny, J. Bennet, K. Benson, J. L. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An Expedited Forwarding PHB (Per-Hop Behavior)," RFC 3246 (Proposed Standard), Internet Engineering Task Force, Mar. 2002. [Online]. Available: <http://www.ietf.org/rfc/rfc3246.txt> 17
- [44] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597 (Proposed Standard), Internet Engineering Task Force, Jun. 1999, updated by RFC 3260. [Online]. Available: <http://www.ietf.org/rfc/rfc2597.txt> 17
- [45] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, Aug. 1993. 18
- [46] D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service," *IEEE/ACM Transactions on Networking*, vol. 6, no. 4, pp. 362–373, Aug. 1998. 18
- [47] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031 (Proposed Standard), Internet Engineering Task Force, Jan. 2001. [Online]. Available: <http://www.ietf.org/rfc/rfc3031.txt> 19
- [48] L. Andersson and R. Asati, "Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field," RFC 5462 (Proposed Standard), Internet Engineering Task Force, Feb. 2009. [Online]. Available: <http://www.ietf.org/rfc/rfc5462.txt> 20
- [49] I. Minei and J. Lucek, *MPLS-Enabled Applications: Emerging Developments and New Technologies*. Wiley, Oct. 2005. 21
- [50] F. L. Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen, "Multi-Protocol Label Switching (MPLS) Support of Differentiated Services," RFC 3270 (Proposed Standard), Internet Engineering Task Force, May 2002, updated by RFC 5462. [Online]. Available: <http://www.ietf.org/rfc/rfc3270.txt> 21

- [51] F. L. Faucheur and W. Lai, "Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering," RFC 3564 (Informational), Internet Engineering Task Force, Jul. 2003, updated by RFC 5462. [Online]. Available: <http://www.ietf.org/rfc/rfc3564.txt> 21
- [52] F. L. Faucheur, "Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering," RFC 4124 (Proposed Standard), Internet Engineering Task Force, Jun. 2005. [Online]. Available: <http://www.ietf.org/rfc/rfc4124.txt> 21
- [53] W. Lai, "Bandwidth Constraints Models for Differentiated Services (Diffserv)-aware MPLS Traffic Engineering: Performance Evaluation," RFC 4128 (Informational), Internet Engineering Task Force, Jun. 2005. [Online]. Available: <http://www.ietf.org/rfc/rfc4128.txt> 22
- [54] I. Bryskin and A. Farrel, "A Lexicography for the Interpretation of Generalized Multiprotocol Label Switching (GMPLS) Terminology within the Context of the ITU-T's Automatically Switched Optical Network (ASON) Architecture," RFC 4397 (Informational), Internet Engineering Task Force, Feb. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4397.txt> 22, 75
- [55] J. Lang, "Link Management Protocol (LMP)," RFC 4204 (Proposed Standard), Internet Engineering Task Force, Oct. 2005. [Online]. Available: <http://www.ietf.org/rfc/rfc4204.txt> 23
- [56] D. Fedyk, O. Aboul-Magd, D. Brungard, J. Lang, and D. Papadimitriou, "A Transport Network View of the Link Management Protocol (LMP)," RFC 4394 (Informational), Internet Engineering Task Force, Feb. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4394.txt> 23
- [57] A. Farrel and I. Bryskin, *GMPLS: Architecture and Applications*. Morgan Kaufmann, Dec. 2005. 23
- [58] J. Moy, "OSPF Version 2," RFC 2328 (Standard), Internet Engineering Task Force, Apr. 1998, updated by RFC 5709. [Online]. Available: <http://www.ietf.org/rfc/rfc2328.txt> 24
- [59] G. Malkin, "RIP Version 2," RFC 2453 (Standard), Internet Engineering Task Force, Nov. 1998, updated by RFC 4822. [Online]. Available: <http://www.ietf.org/rfc/rfc2453.txt> 24
- [60] D. Katz, K. Kompella, and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2," RFC 3630 (Proposed Standard), Internet Engineering Task Force, Sep. 2003, updated by RFCs 4203, 5786. [Online]. Available: <http://www.ietf.org/rfc/rfc3630.txt> 24

REFERENCES

107

- [61] K. Kompella and Y. Rekhter, "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)," RFC 4203 (Proposed Standard), Internet Engineering Task Force, Oct. 2005, updated by RFCs 6001, 6002. [Online]. Available: <http://www.ietf.org/rfc/rfc4203.txt> 24
- [62] L. Berger, I. Bryskin, A. Zinin, and R. Coltun, "The OSPF Opaque LSA Option," RFC 5250 (Proposed Standard), Internet Engineering Task Force, Jul. 2008. [Online]. Available: <http://www.ietf.org/rfc/rfc5250.txt> 24
- [63] G. S. Pavani, L. G. Zuliani, H. Waldman, and M. Magalhães, "Distributed approaches for impairment-aware routing and wavelength assignment algorithms in GMPLS networks," *Computer Networks*, vol. 52, no. 10, pp. 1905–1915, 2008, challenges and Opportunities in Advanced Optical Networking. [Online]. Available: <http://www.sciencedirect.com/science/article/B6VRG-4S21TVC-2/2/28dc967517615db34a6d08ce05a73385> 24, 86
- [64] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209 (Proposed Standard), Internet Engineering Task Force, Dec. 2001, updated by RFCs 3936, 4420, 4874, 5151, 5420, 5711. [Online]. Available: <http://www.ietf.org/rfc/rfc3209.txt> 25
- [65] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions," RFC 3473 (Proposed Standard), Internet Engineering Task Force, Jan. 2003, updated by RFCs 4003, 4201, 4420, 4783, 4874, 4873, 4974, 5063, 5151, 5420, 6002, 6003. [Online]. Available: <http://www.ietf.org/rfc/rfc3473.txt> 25
- [66] A. Satyanarayana and R. Rahman, "Extensions to GMPLS Resource Reservation Protocol (RSVP) Graceful Restart," RFC 5063 (Proposed Standard), Internet Engineering Task Force, Oct. 2007. [Online]. Available: <http://www.ietf.org/rfc/rfc5063.txt> 25
- [67] A. Farrel, A. Ayyangar, and J. Vasseur, "Inter-Domain MPLS and GMPLS Traffic Engineering – Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions," RFC 5151 (Proposed Standard), Internet Engineering Task Force, Feb. 2008. [Online]. Available: <http://www.ietf.org/rfc/rfc5151.txt> 25
- [68] R. Aggarwal, D. Papadimitriou, and S. Yasukawa, "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)," RFC 4875 (Proposed

- Standard), Internet Engineering Task Force, May 2007. [Online]. Available: <http://www.ietf.org/rfc/rfc4875.txt> 25
- [69] S. Yasukawa, “Supporting Multipoint-to-Point Label Switched Paths in Multiprotocol Label Switching Traffic Engineering,” IETF draft, work in progress, Internet Engineering Task Force, Oct. 2009. [Online]. Available: <http://tools.ietf.org/html/draft-yasukawa-mpls-mp2p-rsvp-te-06> 25
- [70] J. Vasseur and J. L. Roux, “Path Computation Element (PCE) Communication Protocol (PCEP),” RFC 5440 (Proposed Standard), Internet Engineering Task Force, Mar. 2009. [Online]. Available: <http://www.ietf.org/rfc/rfc5440.txt> 26
- [71] M. El Barachi, N. Kara, and R. Dssouli, “Towards a service-oriented network virtualization architecture,” in *Kaleidoscope: Beyond the Internet? - Innovations for Future Networks and Services, 2010 ITU-T*, Dec. 2010, pp. 1–7. 27
- [72] D. Colle, B. Jooris, P. Gurzi, M. Pickavet, and P. Demeester, “Network virtualization and programmability,” in *Optoelectronics and Communications Conference (OECC), 2010 15th*, Jul. 2010, pp. 414–415. 28
- [73] K. Oberle, M. Kessler, M. Stein, T. Voith, D. Lamp, and S. Berger, “Network virtualization: The missing piece,” in *Intelligence in Next Generation Networks, 2009. ICIN 2009. 13th International Conference on*, Oct. 2009, pp. 1–6. 28
- [74] “GENI - Global Environment for Network Innovations,” Feb. 2011. [Online]. Available: <http://www.geni.net> 31
- [75] “AKARI - Architecture Design Project for New Generation Network,” Feb. 2011. [Online]. Available: <http://akari-project.nict.go.jp> 31
- [76] “FEDERICA - Federated E-infrastructure Dedicated to European Researchers Innovating in Computing Network Architectures,” Feb. 2011. [Online]. Available: <http://www.fp7-federica.eu> 31
- [77] “PlanetLab - An open platform for developing, deploying, and accessing planetary-scale services,” Feb. 2011. [Online]. Available: <http://www.planet-lab.org> 31
- [78] “GÉANT2 network,” Feb. 2011. [Online]. Available: <http://www.geant2.net> 32

REFERENCES

109

- [79] P. Szegedi, S. Figuerola, M. Campanella, V. Maglaris, and C. Cervelló-Pastor, “With evolution for revolution: managing FEDERICA for future Internet research,” *IEEE Communications Magazine*, vol. 47, no. 7, pp. 34–39, Jul. 2009. 32
- [80] T. Takeda, D. Brungard, D. Papadimitriou, and H. Ould-Brahim, “Layer 1 virtual private networks: driving forces and realization by GMPLS,” *IEEE Communications Magazine*, vol. 43, no. 7, pp. 60–67, Jul. 2005. 36
- [81] A. Bianco, J. Finochietto, M. Mellia, F. Neri, and G. Galante, “Multistage switching architectures for software routers,” *IEEE Network*, vol. 21, no. 4, pp. 15–21, Jul./Aug. 2007. 49
- [82] “TEQUILA project,” Mar. 2011. [Online]. Available: <http://www.ist-tequila.org> 51
- [83] “RSVP-TE daemon for DiffServ over MPLS under Linux,” Mar. 2011. [Online]. Available: <http://dsmpis.atlantis.ugent.be> 51
- [84] “BORA-BORA: Building Open Router Architecture Based On Router Aggregation,” Mar. 2011. [Online]. Available: http://www.ricercailiana.it/prin/dettaglio_completo_prin_en-2005097340.htm 51
- [85] “Quagga Software Routing Suite,” Mar. 2011. [Online]. Available: <http://www.quagga.net> 51
- [86] “XORP,” Mar. 2011. [Online]. Available: <http://www.xorp.org> 51
- [87] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, “Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised),” RFC 4601 (Proposed Standard), Internet Engineering Task Force, Aug. 2006, updated by RFCs 5059, 5796. [Online]. Available: <http://www.ietf.org/rfc/rfc4601.txt> 51
- [88] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, “Internet Group Management Protocol, Version 3,” RFC 3376 (Proposed Standard), Internet Engineering Task Force, Oct. 2002, updated by RFC 4604. [Online]. Available: <http://www.ietf.org/rfc/rfc3376.txt> 51
- [89] R. Vida and L. Costa, “Multicast Listener Discovery Version 2 (MLDv2) for IPv6,” RFC 3810 (Proposed Standard), Internet Engineering Task Force, Jun. 2004, updated by RFC 4604. [Online]. Available: <http://www.ietf.org/rfc/rfc3810.txt> 51
- [90] S. Nadas, “Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6,” RFC 5798 (Proposed Standard), Internet Engineering Task Force, Mar. 2010. [Online]. Available: <http://www.ietf.org/rfc/rfc5798.txt> 51

- [91] “Mikrotik RouterOS,” Mar. 2011. [Online]. Available: <http://www.mikrotik.com> 51
- [92] “Vyatta Core,” Mar. 2011. [Online]. Available: <http://www.vyatta.org> 51
- [93] “Netkit,” Mar. 2011. [Online]. Available: <http://wiki.netkit.org> 52
- [94] “Debian,” Mar. 2011. [Online]. Available: <http://www.debian.org> 52
- [95] “Roma Tre University - Computer Networks Laboratory,” Mar. 2011. [Online]. Available: <http://www.dia.uniroma3.it/~compunet/www/view/group.php?id=compunet> 52
- [96] “User-Mode Linux,” Mar. 2011. [Online]. Available: <http://user-mode-linux.sourceforge.net> 52
- [97] “NetML,” Mar. 2011. [Online]. Available: <http://www.dia.uniroma3.it/~compunet/netml> 52
- [98] “OpenFlow,” Mar. 2011. [Online]. Available: <http://www.openflow.org> 52
- [99] “NetFPGA,” Mar. 2011. [Online]. Available: <http://www.netfpga.org> 52
- [100] “Junos SDK,” Mar. 2011. [Online]. Available: <http://www.juniper.net/us/en/products-services/junos-developer/junos-sdk> 52
- [101] “Cisco AXP,” Mar. 2011. [Online]. Available: <http://developer.cisco.com/web/axp> 52
- [102] “Linux IMQ - Intermediate Queueing Device,” Apr. 2011. [Online]. Available: <http://linuximq.mantech.ro> 53
- [103] “MPLS for Linux project,” Apr. 2011. [Online]. Available: <http://sourceforge.net/projects/mpls-linux/develop> 55
- [104] “iproute2 - Linux Foundation,” Apr. 2011. [Online]. Available: <http://www.linuxfoundation.org/collaborate/workgroups/networking/iproute2> 55
- [105] “netfilter/iptables project,” Apr. 2011. [Online]. Available: <http://www.netfilter.org> 55
- [106] “Linux Ethernet bridge firewall tables,” Apr. 2011. [Online]. Available: <http://ebtables.sourceforge.net> 55
- [107] “HTB Linux kernel implementation,” Feb. 2011. [Online]. Available: <http://luxik.cdi.cz/~devik/qos/htb> 59

REFERENCES

111

- [108] L. G. Zuliani, “Arquitetura e Implementação de um Serviço de Informações Topológicas e de Engenharia de Tráfego para Sistemas RWA,” Master’s Thesis (in Portuguese), Unicamp - FEEC - DCA, Dec. 2006, tutor: Prof. Dr. Mauricio Ferreira Magalhães. [Online]. Available: <http://cutter.unicamp.br/68>
- [109] “Virtual Network User-Mode-Linux (VNUML),” Apr. 2011. [Online]. Available: <http://neweb.dit.upm.es/vnumlwiki/69>
- [110] Y. Lee, G. Bernstein, and W. Imajuku, “Framework for GMPLS and PCE Control of Wavelength Switched Optical Networks (WSON),” IETF draft, work in progress, Internet Engineering Task Force, Feb. 2011. [Online]. Available: <http://tools.ietf.org/html/draft-ietf-ccamp-rwa-wson-framework-12/73>
- [111] Y. Zhai, Y. Pointurier, S. Subramaniam, and M. Brandt-Pearce, “Performance of dedicated path protection in transmission-impaired DWDM networks,” in *Communications, 2007. ICC ’07. IEEE International Conference on*, Jun. 2007, pp. 2342–2347. 73
- [112] Y. Lee, G. Bernstein, D. Li, and G. Martinelli, “A Framework for the Control of Wavelength Switched Optical Networks (WSON) with Impairments,” IETF draft, work in progress, Internet Engineering Task Force, Oct. 2010. [Online]. Available: <http://tools.ietf.org/html/draft-ietf-ccamp-wson-impairments-04/73,76>
- [113] S. Azodolmolky, M. Klinkowski, E. Marin, D. Careglio, J. S. Pareta, and I. Tomkos, “A survey on physical layer impairments aware routing and wavelength assignment algorithms in optical networks,” *Computer Networks*, vol. 53, no. 7, pp. 926–944, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/B6VRG-4V47CMS-3/2/5e371fb273dd5cab455292b46d6faa4e/74,75,76,79>
- [114] G. Markidis and A. Tzanakaki, “Routing and wavelength assignment algorithms in survivable WDM networks under physical layer constraints,” in *Broadband Communications, Networks and Systems, 2008. BROADNETS 2008. 5th International Conference on*, Sep. 2008, pp. 191–196. 74, 76
- [115] A. Askarian, Y. Zhai, S. Subramaniam, Y. Pointurier, and M. Brandt-Pearce, “Protection and restoration from link failures in DWDM networks: A cross-layer study,” in *Communications, 2008. ICC ’08. IEEE International Conference on*, May 2008, pp. 5448–5452. 74, 76

- [116] E. Mannie and D. Papadimitriou, "Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)," RFC 4427 (Informational), Internet Engineering Task Force, Mar. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4427.txt> 75
- [117] D. Papadimitriou and E. Mannie, "Analysis of Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery Mechanisms (including Protection and Restoration)," RFC 4428 (Informational), Internet Engineering Task Force, Mar. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4428.txt> 75
- [118] D. Adami, S. Giordano, M. Pagano, and L. Zuliani, "Lightpath survivability with QoT guarantees: Developing and evaluating a new algorithm," in *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010 International Symposium on*, Jul. 2010, pp. 395–398. 76
- [119] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, pp. 325–336, 1984. [Online]. Available: <http://dx.doi.org/10.1002/net.3230140209> 77, 89
- [120] T. Feng and H. Mouftah, "Implementation issues for asynchronous criticality avoidance protocol in multifiber WDM networks," in *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on*, vol. 2, May 2003, pp. 871–874. 80
- [121] G. S. Pavani and H. Waldman, "Using Genetic Algorithms in Constrained Routing and Wavelength Assignment," in *Proceedings of the 8th IFIP Working Conference on Optical Network Design and Modelling (ONDM'04)*, vol. 1, Feb. 2004, pp. 565–584. 80, 81
- [122] E. M. Varvarigos, V. Sourlas, and K. Christodouloupoulos, "Routing and scheduling connections in networks that support advance reservations," *Computer Networks*, vol. 52, no. 15, pp. 2988–3006, 2008, complex Computer and Communication Networks. [Online]. Available: <http://www.sciencedirect.com/science/article/B6VRG-4SXYG39-1/2/a2bc96336e73919fcf329124decf1053> 84
- [123] M. Ali and J. Deogun, "Power-efficient design of multicast wavelength-routed networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1852–1862, Oct. 2000. 84
- [124] J. Y. Yen, "Finding the K shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971. [Online]. Available: <http://mansci.journal.informs.org/cgi/content/abstract/17/11/712> 86

REFERENCES

113

- [125] D. Lucerna, M. Tornatore, B. Mukherjee, and A. Pattavina, "Dynamic routing of connections with known duration in WDM networks," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, Dec. 2009, pp. 1–7. [92](#)

List of Acronyms

3R	Reamplification, Reshaping and Retiming
AAA	Authentication, Authorization and Accounting
APD	Avalanche PhotoDiode
API	Application Programming Interface
AS	Autonomous System
ASE	Amplified Spontaneous Emission
ASON	Automatic Switched Optical Network
ATED	Alliance Traffic Engineering Database
ATM	Asynchronous Transfer Mode
BE	Best Effort
BGP	Border Gateway Protocol
CAPEX	Capital Expenditure
CLA	Critical Link Avoidance
CoS	Class of Service
CP	Control Plane
CSPF	Constrained Shortest Path First
CT	Class-Type
DP	Data Plane
DRR	Deficit Round Robin
DS	Differentiated Services, better known as DiffServ

DSCP	DiffServ Code Point
DWDM	Dense Wavelength Division Multiplexing
E-LSP	EXP-Inferred-PHB Scheduling Class LSP
EDFA	Erbium Doped Fiber Amplifier
ERO	Explicit Route Object
FAR	Fixed-Alternated Routing
FDM	Frequency Division Multiplexing
FEC	Forwarding Equivalence Class
FIE	Flexible Infrastructure Element
FIFO	First-In First-Out
FR	Fixed-Alternated Routing
FSR	Flexible Software Router
GMPLS	Generalized MultiProtocol Label Switching
GRED	Generalized Random Early Detection
HDLC	High-level Data Link Control
HTB	Hierarchical Token Bucket
IA-RWA	Impairment Aware Routing and Wavelength Assignment
ICANN	Internet Corporation for Assigned Names and Numbers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
ISP	Internet Service Provider
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
L-LSP	Label-Only-Inferred-PHB Scheduling Class LSP
L1	Layer 1
LANE	LAN Emulation

List of Acronyms

117

LCA	LSR Control Agent
LCS	LSR Control System
LED	Light Emitting Diode
LFIB	Label Forwarding Information Base
LIFD	Last-In First-Drop
LLEA	LSR Local Enforcement Agent
LMP	Link Management Protocol
LoL	Loss of Light
LP	Lightpath
LSA	Link State Advertisement
LSDB	Link State DataBase
LSP	Label Switched Path
LSR	Label Switching Router
MAM	Maximum Allocation Model
MCP-RWA	Multipath CLA power-aware RWA
MILP	Mixed-Integer Linear Programming
MLD	Multicast Listener Discovery
MP2P	MultiPoint-to-Point
MPLS	MultiProtocol Label Switching
MPOA	MultiProtocol Over ATM
NGN	Next Generation Network
NGNE	NGN Element
NlaaS	Network Infrastructure as a Service
NMS	Network Management System
NREN	National Research and Education Networks
OADM	Optical Add/Drop Multiplexer

OCS	Optical Circuit Switching
OLT	Optical Line Terminal
OPEX	Operational Expenditure
OPM	Optical Performance Monitoring
OS	Operating System
OSNR	Optical Signal to Noise Ratio
OSPF	Open Shortest Path First
P2MP	Point-to-MultiPoint
PCC	Path Computation Client
PCE	Path Computation Element,
PCEP	Path Computation Element communication Protocol
PHB	Per-Hop Behaviour
PIM-SM	Protocol Independent Multicast - Sparse Mode
PIN	Positive-Intrinsic-Negative
PMD	Polarization Mode Dispersion
POTS	Plain Old Telephone Service
PXC	Photonic Cross-Connect
QoE	Quality of Experience
QoS	Quality of Service
QoT	Quality of Transmission
RDM	Russian Dolls Model
RED	Random Early Detection
RIO	RED with In/Out
RIP	Routing Information Protocol
RWA	Routing and Wavelength Assignment
SLA	Service Level Agreement

List of Acronyms

119

SLIP	Serial Line Internet Protocol
SOHO	Small Office/Home Office
SR	Software Router
SS7	Signaling System No. 7
SSEM	Service Specific Elaboration Module
SSM	Service-Specific module
TDM	Time Division Multiplexing
TE	Traffic Engineering
TED	Traffic Engineering Database
TLV	Type-Length-Value
TNA	Transport Network Alliance
TNVE	Transport Virtual Network Environment
UML	User-Mode Linux
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WDM	Wavelength Division Multiplexing
WFQ	Weighted Fair Queuing
WG	Wavelength Graph
WRPN	Wavelength Routed Photonic Network
XML	eXtensible Markup Language

Appendix A

Manual Configuration of a Software Router DP

This appendix list a `bash` script file that illustrates all sort of commands necessary to manually setup LSPs (i.e., without CP intervention), using the proposed DS-MPLS framework.

set_lsp_Treviso.sh

```
#!/bin/bash

## Filename: Treviso.sh    Version: 1.3
## Zuliani 20100909

## default scheduler tree (DS, CP and BE data)

#HTB root disc
tc qdisc add dev eth1 root handle 1: htb default 99
tc qdisc add dev eth2 root handle 1: htb default 99
tc qdisc add dev eth3 root handle 1: htb default 99

#HTB root class
tc class add dev eth1 parent 1: classid 1:1 htb rate 90Mbit ceil 90Mbit
tc class add dev eth2 parent 1: classid 1:1 htb rate 90Mbit ceil 90Mbit
tc class add dev eth3 parent 1: classid 1:1 htb rate 90Mbit ceil 90Mbit
#Control Plane data class
tc class add dev eth1 parent 1:1 classid 1:12 htb rate 5Mbit ceil 5Mbit
tc class add dev eth2 parent 1:1 classid 1:12 htb rate 5Mbit ceil 5Mbit
tc class add dev eth3 parent 1:1 classid 1:12 htb rate 5Mbit ceil 5Mbit

tc qdisc add dev eth1 parent 1:12 handle 12:0 pfifo
tc qdisc add dev eth2 parent 1:12 handle 12:0 pfifo
tc qdisc add dev eth3 parent 1:12 handle 12:0 pfifo

#icmp (do NOT set this when testing with ping)
tc filter add dev eth1 parent 1:0 protocol 0x8847 \
    pref 2 u32 match u8 0x01 0xff at 13 flowid 1:12
tc filter add dev eth2 parent 1:0 protocol 0x8847 \
```

```
pref 2 u32 match u8 0x01 0xff at 13 flowid 1:12
tc filter add dev eth3 parent 1:0 protocol 0x8847 \
pref 2 u32 match u8 0x01 0xff at 13 flowid 1:12

tc filter add dev eth1 parent 1:0 protocol ip \
pref 3 u32 match u8 0x01 0xff at 9 flowid 1:12
tc filter add dev eth2 parent 1:0 protocol ip \
pref 3 u32 match u8 0x01 0xff at 9 flowid 1:12
tc filter add dev eth3 parent 1:0 protocol ip \
pref 3 u32 match u8 0x01 0xff at 9 flowid 1:12

#ospf
tc filter add dev eth1 parent 1:0 protocol 0x8847 \
pref 4 u32 match u8 0x59 0xff at 13 flowid 1:12
tc filter add dev eth2 parent 1:0 protocol 0x8847 \
pref 4 u32 match u8 0x59 0xff at 13 flowid 1:12
tc filter add dev eth3 parent 1:0 protocol 0x8847 \
pref 4 u32 match u8 0x59 0xff at 13 flowid 1:12

tc filter add dev eth1 parent 1:0 protocol ip \
pref 5 u32 match u8 0x59 0xff at 9 flowid 1:12
tc filter add dev eth2 parent 1:0 protocol ip \
pref 5 u32 match u8 0x59 0xff at 9 flowid 1:12
tc filter add dev eth3 parent 1:0 protocol ip \
pref 5 u32 match u8 0x59 0xff at 9 flowid 1:12

#rsvp-te
tc filter add dev eth1 parent 1:0 protocol 0x8847 \
pref 6 u32 match u8 0x2e 0xff at 13 flowid 1:12
tc filter add dev eth2 parent 1:0 protocol 0x8847 \
pref 6 u32 match u8 0x2e 0xff at 13 flowid 1:12
tc filter add dev eth3 parent 1:0 protocol 0x8847 \
pref 6 u32 match u8 0x2e 0xff at 13 flowid 1:12

tc filter add dev eth1 parent 1:0 protocol ip \
pref 7 u32 match u8 0x2e 0xff at 9 flowid 1:12
tc filter add dev eth2 parent 1:0 protocol ip \
pref 7 u32 match u8 0x2e 0xff at 9 flowid 1:12
tc filter add dev eth3 parent 1:0 protocol ip \
pref 7 u32 match u8 0x2e 0xff at 9 flowid 1:12

#BE traffic class
tc class add dev eth1 parent 1:1 classid 1:99 htb rate 5Mbit ceil 5Mbit
tc class add dev eth2 parent 1:1 classid 1:99 htb rate 5Mbit ceil 5Mbit
tc class add dev eth3 parent 1:1 classid 1:99 htb rate 5Mbit ceil 5Mbit

tc qdisc add dev eth1 parent 1:99 handle 99:0 red limit 5.4MB \
min 200KB max 400KB burst 300 avpkt 1000 bandwidth 5Mbit probability 0.4
tc qdisc add dev eth2 parent 1:99 handle 99:0 red limit 5.4MB \
min 200KB max 400KB burst 300 avpkt 1000 bandwidth 5Mbit probability 0.4
tc qdisc add dev eth3 parent 1:99 handle 99:0 red limit 5.4MB \
min 200KB max 400KB burst 300 avpkt 1000 bandwidth 5Mbit probability 0.4

#LSP-tunneled DS traffic class
tc class add dev eth1 parent 1:1 classid 1:11 htb rate 80Mbit ceil 80Mbit
tc class add dev eth2 parent 1:1 classid 1:11 htb rate 80Mbit ceil 80Mbit
tc class add dev eth3 parent 1:1 classid 1:11 htb rate 80Mbit ceil 80Mbit
tc qdisc add dev eth1 parent 1:11 handle 11: htb
tc qdisc add dev eth2 parent 1:11 handle 11: htb
```

```
tc qdisc add dev eth3 parent 1:11 handle 11: htb

#match any frame that has a shim header
tc filter add dev eth1 parent 1:0 protocol 0x8847 \
    pref 8 u32 match u8 0x00 0x00 at 0 flowid 1:11
tc filter add dev eth2 parent 1:0 protocol 0x8847 \
    pref 8 u32 match u8 0x00 0x00 at 0 flowid 1:11
tc filter add dev eth3 parent 1:0 protocol 0x8847 \
    pref 8 u32 match u8 0x00 0x00 at 0 flowid 1:11

## ***
## LSPs #1      Type: E-LSP                                ##
## upstream:   Granada --> Treviso   outbound IF: ---   Label= --- ##
## downstream: Treviso --> Granada   inbound IF: eth1   Label=13011 ##
## scheduler class: 11:2                                     ##

## E-LSP outbound IF scheduler
tc class add dev eth1 parent 11: classid 11:2 htb rate 30Mbit ceil 30Mbit
tc qdisc add dev eth1 parent 11:2 handle 20: htb

#label matching
tc filter add dev eth1 parent 11:0 protocol 0x8847 \
    pref 9 u32 match u32 0x032d3000 0xffffffff000 at 0 flowid 11:2

#E-LSP
tc class add dev eth1 parent 20: classid 20:1 htb rate 30Mbit ceil 30Mbit

#AF class 1
tc class add dev eth1 parent 20:1 classid 20:10 htb rate 8mbit ceil 30Mbit
tc qdisc add dev eth1 parent 20:10 handle 5: gred setup DPs 8 default 3 prio

#AF Class 1 DP 1
tc qdisc change dev eth1 parent 20:10 gred limit 3.6MB min 150KB max 450KB \
    burst 250 avpkt 1000 bandwidth 10Mbit DP 2 probability 0.01 prio 2
tc filter add dev eth1 parent 20:0 protocol 0x8847 \
    pref 2 handle 0x02 tcindex classid 20:10

#AF Class 1 DP 2
tc qdisc change dev eth1 parent 20:10 gred limit 3.6MB min 150KB max 450KB \
    burst 250 avpkt 1000 bandwidth 10Mbit DP 3 probability 0.02 prio 3
tc filter add dev eth1 parent 20:0 protocol 0x8847 \
    pref 2 handle 0x03 tcindex classid 20:10

#AF class 2
tc class add dev eth1 parent 20:1 classid 20:20 htb rate 8Mbit ceil 30Mbit
tc qdisc add dev eth1 parent 20:20 handle 6: gred setup DPs 8 default 5 prio

#AF Class 2 DP 1
tc qdisc change dev eth1 parent 20:20 gred limit 2.4MB min 100KB max 300KB \
    burst 175 avpkt 1000 bandwidth 10Mbit DP 4 probability 0.02 prio 2
tc filter add dev eth1 parent 20:0 protocol 0x8847 \
    pref 2 handle 0x04 tcindex classid 20:20

#AF Class 2 DP 2
tc qdisc change dev eth1 parent 20:20 gred limit 2.4MB min 100KB max 300KB \
    burst 175 avpkt 1000 bandwidth 10Mbit DP 5 probability 0.04 prio 3
tc filter add dev eth1 parent 20:0 protocol 0x8847 \
    pref 2 handle 0x05 tcindex classid 20:20
```

```
#AF Class 3
tc class add dev eth1 parent 20:1 classid 20:30 htb rate 8Mbit ceil 30Mbit
tc qdisc add dev eth1 parent 20:30 handle 7: gred setup DPs 8 default 7 prio

#AF Class 3 DP 1
tc qdisc change dev eth1 parent 20:30 gred limit 1.6MB min 65KB max 200KB \
    burst 125 avpkt 1000 bandwidth 5Mbit DP 6 probability 0.03 prio 2
tc filter add dev eth1 parent 20:0 protocol 0x8847 \
    pref 2 handle 0x06 tcindex classid 20:30

#AF Class 3 DP 2
tc qdisc change dev eth1 parent 20:30 gred limit 1.6MB min 65KB max 200KB \
    burst 125 avpkt 1000 bandwidth 5Mbit DP 7 probability 0.05 prio 3
tc filter add dev eth1 parent 20:0 protocol 0x8847 \
    pref 2 handle 0x07 tcindex classid 20:30

# EF
tc class add dev eth1 parent 20:1 classid 20:50 htb rate 5mbit ceil 5Mbit
tc qdisc add dev eth1 parent 20:50 handle 9: pfifo limit 10
tc filter add dev eth1 parent 20:0 protocol 0x8847 \
    pref 2 handle 0x01 tcindex classid 20:50

#E-LSP ## BE
tc class add dev eth1 parent 20:1 classid 20:40 htb rate 1Mbit ceil 30Mbit
tc qdisc add dev eth1 parent 20:40 handle 8: pfifo
tc filter add dev eth1 parent 20:0 protocol 0x8847 \
    pref 2 handle 0x00 tcindex classid 20:40

## E-LSP packet forwarding configuration
mpls labelspace set dev eth1 labelspace 0
#Downstream (granada->Treviso)
mpls ilm add label gen 12110 labelspace 0
#upstream (Treviso->granada)
NHLFE_CMD1='mpls nhlfe add key 0 instructions \
    ds2exp 0x3f 0x00 0 0x2e 1 0x0a 2 0x0c 3 0x12 4 0x14 5 0x1a 6 0x1c 7 \
    exp2tc 1 0x01 2 0x02 3 0x03 4 0x04 5 0x05 6 0x06 7 0x07 \
    push gen 13011 nexthop eth1 ipv4 172.16.13.2'
NHLFE_KEY1='echo $NHLFE_CMD1 | awk '{print $4}''

# Only core DS_LSR:
#mpls xc add ilm_label gen <LABEL> ilm_labelspace 0 nhlfe_key $NHLFE_KEY

# Classifying incoming (or generated) data to tunneled
iptables -t mangle -A OUTPUT -m dscp \
    --dscp-class AF11 -j MARK --set-mark 101
iptables -t mangle -A OUTPUT -m dscp --dscp-class AF11 -j ACCEPT
iptables -t mangle -A OUTPUT -m dscp \
    --dscp-class AF12 -j MARK --set-mark 101
iptables -t mangle -A OUTPUT -m dscp --dscp-class AF12 -j ACCEPT
iptables -t mangle -A OUTPUT -m dscp \
    --dscp-class AF21 -j MARK --set-mark 101
iptables -t mangle -A OUTPUT -m dscp --dscp-class AF21 -j ACCEPT
iptables -t mangle -A OUTPUT -m dscp \
    --dscp-class AF22 -j MARK --set-mark 101
iptables -t mangle -A OUTPUT -m dscp --dscp-class AF22 -j ACCEPT
iptables -t mangle -A OUTPUT -m dscp \
    --dscp-class AF31 -j MARK --set-mark 101
```

```

iptables -t mangle -A OUTPUT -m dscp --dscp-class AF31 -j ACCEPT
iptables -t mangle -A OUTPUT -m dscp \
    --dscp-class AF32 -j MARK --set-mark 101
iptables -t mangle -A OUTPUT -m dscp --dscp-class AF32 -j ACCEPT

iptables -t mangle -A PREROUTING -m dscp \
    --dscp-class AF11 -j MARK --set-mark 101
iptables -t mangle -A PREROUTING -m dscp --dscp-class AF11 -j ACCEPT
iptables -t mangle -A PREROUTING -m dscp \
    --dscp-class AF12 -j MARK --set-mark 101
iptables -t mangle -A PREROUTING -m dscp --dscp-class AF12 -j ACCEPT
iptables -t mangle -A PREROUTING -m dscp \
    --dscp-class AF21 -j MARK --set-mark 101
iptables -t mangle -A PREROUTING -m dscp --dscp-class AF21 -j ACCEPT
iptables -t mangle -A PREROUTING -m dscp \
    --dscp-class AF22 -j MARK --set-mark 101
iptables -t mangle -A PREROUTING -m dscp --dscp-class AF22 -j ACCEPT
iptables -t mangle -A PREROUTING -m dscp \
    --dscp-class AF31 -j MARK --set-mark 101
iptables -t mangle -A PREROUTING -m dscp --dscp-class AF31 -j ACCEPT
iptables -t mangle -A PREROUTING -m dscp \
    --dscp-class AF32 -j MARK --set-mark 101
iptables -t mangle -A PREROUTING -m dscp --dscp-class AF32 -j ACCEPT

## Routing tables
ip rule add table 99 prio 30599
ip rule add fwmark 101 table 101 prio 30101
ip route add default via 172.16.13.2 dev eth1 mpls $NHLFE_KEY1 table 101

## ***
## LSPs #2      Type: L-LSP (AF4x)
## upstream:   Granada --> Treviso   outbound IF: --- Label= ---
## downstream: Treviso --> Granada   inbound IF: eth3 Label=13231
## scheduler class: 11:4

## L-LSP (AF4x) outbound IF scheduler
tc class add dev eth3 parent 11: classid 11:4 htb rate 20Mbit ceil 20Mbit
tc qdisc add dev eth3 parent 11:4 handle 40: htb
tc class add dev eth3 parent 40: classid 40:1 htb rate 20Mbit ceil 20Mbit
tc qdisc add dev eth3 parent 40:1 handle 41: gred setup DPs 4 default 3 prio

#label matching
tc filter add dev eth3 parent 11:0 protocol 0x8847 \
    pref 10 u32 match u32 0x033af000 0xffffffff000 at 0 flowid 11:4

#AF Class 4 DP 1
tc qdisc change dev eth3 parent 40:1 gred limit 1.0MB min 45KB max 100KB \
    burst 80 avpkt 1000 bandwidth 20Mbit DP 1 probability 0.03 prio 1
tc filter add dev eth3 parent 40:0 protocol 0x8847 \
    pref 1 handle 0x01 tcindex classid 40:1

#AF Class 4 DP 2
tc qdisc change dev eth3 parent 40:1 gred limit 1.0MB min 45KB max 100KB \
    burst 80 avpkt 1000 bandwidth 20Mbit DP 2 probability 0.05 prio 2
tc filter add dev eth3 parent 40:0 protocol 0x8847 \
    pref 1 handle 0x02 tcindex classid 40:1

```

126

A. Manual Configuration of a Software Router DP

```
#AF Class 4 DP 3
tc qdisc change dev eth3 parent 40:1 gred limit 1.0MB min 45KB max 100KB \
    burst 80 avpkt 1000 bandwidth 20Mbit DP 3 probability 0.07 prio 3
tc filter add dev eth3 parent 40:0 protocol 0x8847 \
    pref 1 handle 0x03 tcindex classid 40:1

## L-LSP (AF4x) packet forwarding configuration
mpls labelspace set dev eth3 labelspace 0
#Downstream (granada->Treviso)
mpls ilm add label gen 14130 labelspace 0
#upstream (Treviso->granada)
NHLFE_CMD3=`mpls nhlfe add key 0 instructions \
    ds2exp 0x3f 0x00 0 0x2e 0 0x0a 1 0x0c 2 0x0e 3 0x12 1 0x14 2 0x16 3 \
    0x1a 1 0x1c 2 0x1e 3 0x22 1 0x24 2 0x26 3 \
    exp2tc 1 0x01 2 0x02 3 0x03 \
    push gen 13231 nexthop eth3 ipv4 172.16.15.4`
NHLFE_KEY3=`echo $NHLFE_CMD3 | awk '{print $4}'`

# Only core DS_LSR:
#mpls xc add ilm_label gen <LABEL> ilm_labelspace 0 nhlfe_key $NHLFE_KEY

## Classifying incoming (or generated) data to tunneled
iptables -t mangle -A OUTPUT -m dscp \
    --dscp-class AF41 -j MARK --set-mark 102
iptables -t mangle -A OUTPUT -m dscp --dscp-class AF41 -j ACCEPT
iptables -t mangle -A OUTPUT -m dscp \
    --dscp-class AF42 -j MARK --set-mark 102
iptables -t mangle -A OUTPUT -m dscp --dscp-class AF42 -j ACCEPT
iptables -t mangle -A OUTPUT -m dscp \
    --dscp-class AF43 -j MARK --set-mark 102
iptables -t mangle -A OUTPUT -m dscp --dscp-class AF43 -j ACCEPT

iptables -t mangle -A PREROUTING -m dscp \
    --dscp-class AF41 -j MARK --set-mark 102
iptables -t mangle -A PREROUTING -m dscp --dscp-class AF41 -j ACCEPT
iptables -t mangle -A PREROUTING -m dscp \
    --dscp-class AF42 -j MARK --set-mark 102
iptables -t mangle -A PREROUTING -m dscp --dscp-class AF42 -j ACCEPT
iptables -t mangle -A PREROUTING -m dscp \
    --dscp-class AF43 -j MARK --set-mark 102
iptables -t mangle -A PREROUTING -m dscp --dscp-class AF43 -j ACCEPT

## Routing tables
#ip rule add table 99 prio 30599
ip rule add fwmark 102 table 102 prio 30102
ip route add default via 172.16.15.4 dev eth3 mpls $NHLFE_KEY3 table 102

## ***
## LSPs #3      Type: L-LSP (EF)                                     ##
## upstream:   Granada --> Treviso   outbound IF: --- Label= ---   ##
## downstream: Treviso --> Granada   inbound IF: eth2 Label=13121  ##
## scheduler class: 11:3                                              ##

## L-LSP (EF) outbound IF scheduler
tc class add dev eth2 parent 11: classid 11:3 htb rate 5Mbit ceil 5Mbit
```



```
tc qdisc add dev eth2 parent 11:3 handle 30: pfifo limit 10

#label matching
tc filter add dev eth2 parent 11:0 protocol 0x8847 \
    pref 9 u32 match u32 0x03341000 0xfffff000 at 0 flowid 11:3

## L-LSP (EF) packet forwarding configuration
mpls labelspace set dev eth2 labelspace 0
#Downstream (granada->Treviso)
mpls ilm add label gen 15120 labelspace 0
#upstream (Treviso->granada)
NHLFE_CMD2=`mpls nhlfe add key 0 instructions \
    ds2exp 0x3f 0x00 0 0x2e 0 0x0a 1 0x0c 2 0x0e 3 0x12 1 0x14 2 0x16 3 \
    0x1a 1 0x1c 2 0x1e 3 0x22 1 0x24 2 0x26 3 \
    exp2tc 1 0x01 2 0x02 3 0x03 \
    push gen 13121 nexthop eth2 ipv4 172.16.14.5`
NHLFE_KEY2=`echo $NHLFE_CMD2 | awk '{print $4}'`

# Only core DS_LSR:
#mpls xc add ilm_label gen <LABEL> ilm_labelspace 0 nhlfe_key $NHLFE_KEY

## Classifying incoming (or generated) data to tunneled
iptables -t mangle -A OUTPUT -m dscp \
    --dscp-class EF -j MARK --set-mark 103
iptables -t mangle -A OUTPUT -m dscp --dscp-class EF -j ACCEPT

iptables -t mangle -A PREROUTING -m dscp \
    --dscp-class EF -j MARK --set-mark 103
iptables -t mangle -A PREROUTING -m dscp --dscp-class EF -j ACCEPT

## Routing tables
#ip rule add table 99 prio 30599
ip rule add fwmark 103 table 103 prio 30103
ip route add default via 172.16.14.5 dev eth2 mpls $NHLFE_KEY2 table 103
```


Appendix B

TED Description

This appendix contains two XML files that exemplify the TED used by both versions of the CP, as part of the DS-MPLS framework.

lsp_topology.xml

LSP (or virtual) topology, fully describes all LSPs currently installed in the network.

ip_topology.xml

IP (or physical) topology, detailed description of routers and IP links that composes the network. Not only nominal characteristics are presented, but also the current status of routers and links.

lsp_topology.xml

```
<lsp_topology>
  <lsp ID="1" FWMARK="17093" TABLE="12189">
    <bw>12500000</bw>
    <delay>20.000000</delay>
    <monetary_cost>25.000000</monetary_cost>
    <type>1</type>
    <priority>1</priority>
    <path>
      <hop LSR="172.16.13.3" IF="172.16.15.3" LABEL="14375"/>
      <hop LSR="172.16.15.4" IF="172.16.16.4" LABEL="262"/>
      <hop LSR="172.16.16.6" IF="-" LABEL="-"/>
    </path>
  </lsp>
  <lsp ID="5" FWMARK="9709" TABLE="234">
    <bw>12500000</bw>
    <delay>10.000000</delay>
    <monetary_cost>12.500000</monetary_cost>
    <type>3</type>
    <priority>1</priority>
    <path>
      <hop LSR="172.16.10.1" IF="172.16.11.1" LABEL="6246"/>
    </path>
  </lsp>
</lsp_topology>
```

```
<hop LSR="172.16.11.5" IF="-" LABEL="-"/>
</path>
</lsp>
<lsp ID="6" FWMARK="9710" TABLE="235">
  <bw>12500000</bw>
  <delay>10.000000</delay>
  <monetary_cost>12.500000</monetary_cost>
  <type>3</type>
  <priority>1</priority>
  <path>
    <hop LSR="172.16.11.5" IF="172.16.11.5" LABEL="14423"/>
    <hop LSR="172.16.10.1" IF="-" LABEL="-"/>
  </path>
</lsp>
<lsp ID="3" FWMARK="1667" TABLE="10149">
  <bw>125000</bw>
  <delay>38.000000</delay>
  <monetary_cost>0.625000</monetary_cost>
  <type>12</type>
  <priority>2</priority>
  <path>
    <hop LSR="172.16.10.2" IF="172.16.13.2" LABEL="13090"/>
    <hop LSR="172.16.13.3" IF="172.16.14.3" LABEL="31746"/>
    <hop LSR="172.16.11.5" IF="172.16.17.5" LABEL="17633"/>
    <hop LSR="172.16.16.6" IF="172.16.16.6" LABEL="3521"/>
    <hop LSR="172.16.15.4" IF="-" LABEL="-"/>
  </path>
</lsp>
<lsp ID="4" FWMARK="1668" TABLE="10150">
  <bw>125000</bw>
  <delay>18.000000</delay>
  <monetary_cost>0.375000</monetary_cost>
  <type>12</type>
  <priority>2</priority>
  <path>
    <hop LSR="172.16.15.4" IF="172.16.15.4" LABEL="32171"/>
    <hop LSR="172.16.13.3" IF="172.16.13.3" LABEL="18059"/>
    <hop LSR="172.16.10.2" IF="-" LABEL="-"/>
  </path>
</lsp>
<lsp ID="7" FWMARK="12029" TABLE="16602">
  <bw>2500000</bw>
  <delay>30.000000</delay>
  <monetary_cost>7.500000</monetary_cost>
  <type>8</type>
  <priority>1</priority>
  <path>
    <hop LSR="172.16.12.7" IF="172.16.18.7" LABEL="7156"/>
    <hop LSR="172.16.16.6" IF="172.16.16.6" LABEL="25812"/>
    <hop LSR="172.16.15.4" IF="172.16.15.4" LABEL="11699"/>
    <hop LSR="172.16.13.3" IF="-" LABEL="-"/>
  </path>
</lsp>
<lsp ID="8" FWMARK="12030" TABLE="16603">
  <bw>2500000</bw>
  <delay>30.000000</delay>
  <monetary_cost>7.500000</monetary_cost>
  <type>8</type>
  <priority>1</priority>
```

```

<path>
  <hop LSR="172.16.13.3" IF="172.16.14.3" LABEL="22602"/>
  <hop LSR="172.16.11.5" IF="172.16.17.5" LABEL="8490"/>
  <hop LSR="172.16.16.6" IF="172.16.18.6" LABEL="27145"/>
  <hop LSR="172.16.12.7" IF="-" LABEL="-"/>
</path>
</lsp>
<lsp ID="9" FWMARK="2286" TABLE="24979">
  <bw>2500000</bw>
  <delay>24.000000</delay>
  <monetary_cost>20.000000</monetary_cost>
  <type>8</type>
  <priority>1</priority>
  <path>
    <hop LSR="172.16.12.7" IF="172.16.12.7" LABEL="26458"/>
    <hop LSR="172.16.10.1" IF="172.16.10.1" LABEL="12346"/>
    <hop LSR="172.16.10.2" IF="172.16.13.2" LABEL="31001"/>
    <hop LSR="172.16.13.3" IF="-" LABEL="-"/>
  </path>
</lsp>
<lsp ID="10" FWMARK="2287" TABLE="24980">
  <bw>2500000</bw>
  <delay>24.000000</delay>
  <monetary_cost>20.000000</monetary_cost>
  <type>8</type>
  <priority>1</priority>
  <path>
    <hop LSR="172.16.13.3" IF="172.16.13.3" LABEL="9136"/>
    <hop LSR="172.16.10.2" IF="172.16.10.2" LABEL="27792"/>
    <hop LSR="172.16.10.1" IF="172.16.12.1" LABEL="13680"/>
    <hop LSR="172.16.12.7" IF="-" LABEL="-"/>
  </path>
</lsp>
</lsp_topology>

```

ip_topology.xml

```

<!-- RFC 3630, RFC 4124, RFC 4125, RFC 4126, RFC 4127, RFC 4128 -->
<!-- and DS-MPLS framework custom extensions -->
<!-- Zuliani 20091212 -->
<ip_topology>
  <lsp>
    <!-- Router Address (top-level TLV) type 1, length 4 octets-->
    <router_address>172.16.10.2</router_address>
    <links>
      <!-- Link (top-level TLV) type 2, length variable -->
      <link ID="1">
        <!--
          Link Type (sub-TLV of Link TLV)
          type 1, length 1 octet. Mandatory
          values: 1 - Point-to-point, 2 - Multi-access
        -->
        <link_type>1</link_type>
        <!--
          Link ID (sub-TLV of Link TLV)
          type 2, length 4 octets. Mandatory
          For point-to-point links,

```

```

    this is the Router ID of the neighbor
    For multi-access links,
    this is the interface address of the designated router
-->
<link_id>172.16.10.1</link_id>
<!--
    Local Interface IP Address (sub-TLV of Link TLV)
    type 3, length N * 4 octets
-->
<local_if>172.16.10.2</local_if>
<!--
    Remote Interface IP Address (sub-TLV of Link TLV)
    type 4, length N * 4 octets
-->
<remote_if>172.16.10.1</remote_if>
<!--
    Traffic engineering metric (sub-TLV of Link TLV)
    type 5, length 4 octets
-->
<te_metric>67341933</te_metric>
<!--
    Maximum bandwidth (sub-TLV of Link TLV)
    type 6, length 4 octets
    True link capacity, IEEE fpf, B/s
-->
<max_bw>12500000</max_bw>
<!--
    Maximum Reservable Bandwidth (sub-TLV of Link TLV)
    type 7, length 4 octets
    may be greater than the maximum bandwidth
    (oversubscription), IEEE fpf, B/s
    the default value should be the Maximum Bandwidth
    Diffserv-aware MPLS Traffic Engineering (DS-TE):
    it MUST now be interpreted as the aggregate
    bandwidth constraint across all Class-Types
-->
<max_res_bw>12500000</max_res_bw>
<!--
    Unreserved bandwidth (sub-TLV of Link TLV)
    type 8, length 32 octets
    bandwidth not yet reserved at each of
    the eight priority levels, IEEE fpf, B/s
    The initial values are all set to the
    Maximum Reservable Bandwidth
    Diffserv-aware MPLS Traffic Engineering (DS-TE):
    it now specifies the amount of bandwidth
    not yet reserved for each of the eight TE-Classes
-->
<unres_bw>
  <prio_0>10000000</prio_0>
  <prio_1>10000000</prio_1>
  <prio_2>10000000</prio_2>
  <prio_3>10000000</prio_3>
  <prio_4>10000000</prio_4>
  <prio_5>10000000</prio_5>
  <prio_6>10000000</prio_6>
  <prio_7>10000000</prio_7>
</unres_bw>
<!--
```

```
Administrative group (sub-TLV of Link TLV)
type 9, length 4 octets bit mask
The Admin Group is also called Resource Class/Color
-->
<admin_group>0x0001</admin_group>
<!--
Bandwidth Constraints - DS-TE (sub-TLV of Link TLV)
type 17, length:
Bandwidth Constraints Model ID (1 octet):
BCM currently in use by LSR. Values:
0 to RDM (Russian Dolls Model)
1 to MAM (Maximum Allocation Model)
2 to MAR (Maximum Allocation with Reservation Model)
Reserved (3 octets)
Bandwidth Constraints (N x 4 octets): IEEE fpf, B/s
-->
<bw_constraints>
  <bcm_id/>
  <bc>
    <bc0>12500000</bc0>
    <bc1>11000000</bc1>
    <bc2>10000000</bc2>
    <bc3>90000000</bc3>
    <bc4>80000000</bc4>
    <bc5>70000000</bc5>
    <bc6>60000000</bc6>
    <bc7>50000000</bc7>
  </bc>
</bw_constraints>
<!--
Link delay (Experimental sub-TLV of Link TLV)
type 32768, length 4 octets
value: IEEE fpf, ms
-->
<delay>8</delay>
<!--
Link monetary cost (Experimental sub-TLV of Link TLV)
type 32769, length 4 octets
value: IEEE fpf, euro/B
-->
<monetary_cost>0.000003</monetary_cost>
</link>
<link ID="2">
  <link_type>1</link_type>
  <link_id>172.16.13.3</link_id>
  <local_if>172.16.13.2</local_if>
  <remote_if>172.16.13.3</remote_if>
  <te_metric>67341933</te_metric>
  <max_bw>12500000</max_bw>
  <max_res_bw>12500000</max_res_bw>
  <unres_bw>
    <prio_0>9875000</prio_0>
    <prio_1>9875000</prio_1>
    <prio_2>9875000</prio_2>
    <prio_3>9875000</prio_3>
    <prio_4>9875000</prio_4>
    <prio_5>9875000</prio_5>
    <prio_6>9875000</prio_6>
    <prio_7>9875000</prio_7>
```

```
</unres_bw>
<admin_group>0x0001</admin_group>
<bw_constraints>
  <bcm_id/>
  <bc>
    <bc0>12500000</bc0>
    <bc1>11000000</bc1>
    <bc2>10000000</bc2>
    <bc3>9000000</bc3>
    <bc4>8000000</bc4>
    <bc5>7000000</bc5>
    <bc6>6000000</bc6>
    <bc7>5000000</bc7>
  </bc>
</bw_constraints>
<delay>8</delay>
<monetary_cost>0.000002</monetary_cost>
</link>
</links>
</lsr>
<lsr>
  <router_address>172.16.10.1</router_address>
  <links>
    <link ID="3">
      <link_type>1</link_type>
      <link_id>172.16.10.2</link_id>
      <local_if>172.16.10.1</local_if>
      <remote_if>172.16.10.2</remote_if>
      <te_metric>67341933</te_metric>
      <max_bw>12500000</max_bw>
      <max_res_bw>12500000</max_res_bw>
      <unres_bw>
        <prio_0>10000000</prio_0>
        <prio_1>10000000</prio_1>
        <prio_2>10000000</prio_2>
        <prio_3>10000000</prio_3>
        <prio_4>10000000</prio_4>
        <prio_5>10000000</prio_5>
        <prio_6>10000000</prio_6>
        <prio_7>10000000</prio_7>
      </unres_bw>
      <admin_group>0x0001</admin_group>
      <bw_constraints>
        <bcm_id/>
        <bc>
          <bc0>12500000</bc0>
          <bc1>11000000</bc1>
          <bc2>10000000</bc2>
          <bc3>9000000</bc3>
          <bc4>8000000</bc4>
          <bc5>7000000</bc5>
          <bc6>6000000</bc6>
          <bc7>5000000</bc7>
        </bc>
      </bw_constraints>
      <delay>8</delay>
      <monetary_cost>0.000003</monetary_cost>
    </link>
    <link ID="4">
```



```
<link_type>1</link_type>
<link_id>172.16.11.5</link_id>
<local_if>172.16.11.1</local_if>
<remote_if>172.16.11.5</remote_if>
<te_metric>67341933</te_metric>
<max_bw>12500000</max_bw>
<max_res_bw>12500000</max_res_bw>
<unres_bw>
  <prio_0>0</prio_0>
  <prio_1>0</prio_1>
  <prio_2>0</prio_2>
  <prio_3>0</prio_3>
  <prio_4>0</prio_4>
  <prio_5>0</prio_5>
  <prio_6>0</prio_6>
  <prio_7>0</prio_7>
</unres_bw>
<admin_group>0x0001</admin_group>
<bw_constraints>
  <bcm_id/>
  <bc>
    <bc0>12500000</bc0>
    <bc1>11000000</bc1>
    <bc2>10000000</bc2>
    <bc3>90000000</bc3>
    <bc4>80000000</bc4>
    <bc5>70000000</bc5>
    <bc6>60000000</bc6>
    <bc7>50000000</bc7>
  </bc>
</bw_constraints>
<delay>10</delay>
<monetary_cost>0.000001</monetary_cost>
</link>
<link ID="5">
  <link_type>1</link_type>
  <link_id>172.16.12.7</link_id>
  <local_if>172.16.12.1</local_if>
  <remote_if>172.16.12.7</remote_if>
  <te_metric>67341933</te_metric>
  <max_bw>12500000</max_bw>
  <max_res_bw>12500000</max_res_bw>
  <unres_bw>
    <prio_0>10000000</prio_0>
    <prio_1>10000000</prio_1>
    <prio_2>10000000</prio_2>
    <prio_3>10000000</prio_3>
    <prio_4>10000000</prio_4>
    <prio_5>10000000</prio_5>
    <prio_6>10000000</prio_6>
    <prio_7>10000000</prio_7>
  </unres_bw>
  <admin_group>0x0001</admin_group>
  <bw_constraints>
    <bcm_id/>
    <bc>
      <bc0>12500000</bc0>
      <bc1>11000000</bc1>
      <bc2>10000000</bc2>
```

```
<bc3>9000000</bc3>
<bc4>8000000</bc4>
<bc5>7000000</bc5>
<bc6>6000000</bc6>
<bc7>5000000</bc7>
</bc>
</bw_constraints>
<delay>8</delay>
<monetary_cost>0.000003</monetary_cost>
</link>
</links>
</lsr>
</ip_topology>
```

Appendix C

VNT Standard Topology Configuration

This appendix list the XML file used with VNUML to launch the standard topology in the VNT live distribution.

VNT_topology.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE vnuml SYSTEM "/usr/share/xml/vnuml/vnuml.dtd">

<!-- DS-MPLS Virtual Network Testbed -->
<!-- Zuliani - 20101228 -->

<vnuml>
  <global>
    <version>1.8</version>
    <simulation_name>dsmpls-vnt</simulation_name>
    <ssh_version>2</ssh_version>
    <ssh_key>/root/.ssh/id_rsa.pub</ssh_key>
    <automac/>
    <!-- <netconfig stp="on" promisc="on" /> -->
    <vm_mgmt type="none" />

    <vm_defaults>
      <filesystem type="cow">
        /root/dsmpls-vnt/fs-dsmpls-vnt-20101226.img
      </filesystem>
      <!--Do NOT use more than 256M if host has less than 2G of RAM -->
      <mem>256M</mem>
      <kernel>/root/dsmpls-vnt/linux-dsmpls-vnt</kernel>
      <shell>/bin/bash</shell>
      <console id="1">xterm</console>
      <forwarding type="ip"/>
    </vm_defaults>
  </global>
```

```
<!-- DP links. The "virtual_bridge" mode allows -->
<!-- config of VMs interfaces with network schedulers -->
<net name="link1" mode="virtual_bridge" />
<net name="link2" mode="virtual_bridge" />
<net name="link3" mode="virtual_bridge" />
<net name="link4" mode="virtual_bridge" />
<net name="link5" mode="virtual_bridge" />
<net name="link6" mode="virtual_bridge" />
<net name="link7" mode="virtual_bridge" />
<net name="link8" mode="virtual_bridge" />
<net name="link9" mode="virtual_bridge" />

<!-- Control Plane switch -->
<!-- <net name="cp_net" mode="uml_switch" /> -->
<net name="cp_net" mode="virtual_bridge" />
<!-- Exterior network -->
<!-- <net name="ext_net" mode="uml_switch" /> -->
<net name="ext_net" mode="virtual_bridge" />

<vm name="granada" order="1">
  <xterm>xterm,-T granada,-e</xterm>
  <if id="1" net="link1">
    <ipv4>172.16.12.7/24</ipv4>
  </if>
  <if id="2" net="link7">
    <ipv4>172.16.18.7/24</ipv4>
  </if>
  <if id="3" net="cp_net">
    <ipv4>10.10.10.7/24</ipv4>
  </if>
  <if id="0" net="ext_net">
    <ipv4>192.168.220.112/22</ipv4>
  </if>
  <route type="ipv4" gw="192.168.220.254">default</route>
</vm>

<vm name="dakar" order="4">
  <xterm>xterm,-T dakar,-e</xterm>
  <if id="2" net="link1">
    <ipv4>172.16.12.1/24</ipv4>
  </if>
  <if id="0" net="link2">
    <ipv4>172.16.10.1/24</ipv4>
  </if>
  <if id="1" net="link8">
    <ipv4>172.16.11.1/24</ipv4>
  </if>
  <if id="3" net="cp_net">
    <ipv4>10.10.10.1/24</ipv4>
  </if>
  <route type="ipv4" gw="172.16.12.7">default</route>
</vm>

<vm name="odessa" order="5">
  <xterm>xterm,-T odessa,-e</xterm>
  <if id="0" net="link2">
    <ipv4>172.16.10.2/24</ipv4>
  </if>
  <if id="1" net="link3">
```

```
<ipv4>172.16.13.2/24</ipv4>
</if>
<if id="2" net="cp_net">
  <ipv4>10.10.10.2/24</ipv4>
</if>
<route type="ipv4" gw="172.16.13.3">default</route>
</vm>

<vm name="treviso" order="3">
  <xterm>xterm,-T treviso,-e</xterm>
  <if id="1" net="link3">
    <ipv4>172.16.13.3/24</ipv4>
  </if>
  <if id="2" net="link9">
    <ipv4>172.16.14.3/24</ipv4>
  </if>
  <if id="3" net="link4">
    <ipv4>172.16.15.3/24</ipv4>
  </if>
  <if id="4" net="cp_net">
    <ipv4>10.10.10.3/24</ipv4>
  </if>
  <if id="0" net="ext_net">
    <ipv4>192.168.220.150/22</ipv4>
  </if>
  <route type="ipv4" gw="192.168.220.254">default</route>
</vm>

<vm name="cairo" order="2">
  <xterm>xterm,-T cairo,-e</xterm>
  <if id="1" net="link4">
    <ipv4>172.16.15.4/24</ipv4>
  </if>
  <if id="2" net="link6">
    <ipv4>172.16.16.4/24</ipv4>
  </if>
  <if id="3" net="cp_net">
    <ipv4>10.10.10.4/24</ipv4>
  </if>
  <if id="0" net="ext_net">
    <ipv4>192.168.220.226/22</ipv4>
  </if>
  <route type="ipv4" gw="192.168.220.254">default</route>
</vm>

<vm name="oslo" order="6">
  <xterm>xterm,-T oslo,-e</xterm>
  <if id="0" net="link6">
    <ipv4>172.16.16.6/24</ipv4>
  </if>
  <if id="1" net="link5">
    <ipv4>172.16.17.6/24</ipv4>
  </if>
  <if id="2" net="link7">
    <ipv4>172.16.18.6/24</ipv4>
  </if>
  <if id="3" net="cp_net">
    <ipv4>10.10.10.6/24</ipv4>
  </if>
```

```
<route type="ipv4" gw="172.16.16.4">default</route>
</vm>

<vm name="beirut" order="7">
  <xterm>xterm,-T beirut,-e</xterm>
  <if id="1" net="link9">
    <ipv4>172.16.14.5/24</ipv4>
  </if>
  <if id="2" net="link5">
    <ipv4>172.16.17.5/24</ipv4>
  </if>
  <if id="0" net="link8">
    <ipv4>172.16.11.5/24</ipv4>
  </if>
  <if id="3" net="cp_net">
    <ipv4>10.10.10.5/24</ipv4>
  </if>
  <route type="ipv4" gw="172.16.14.3">default</route>
</vm>

<!-- Host settings -->
<host>
  <hostif net="ext_net">
    <ipv4>192.168.220.254/22</ipv4>
  </hostif>
  <hostif net="cp_net">
    <ipv4>10.10.10.254/24</ipv4>
  </hostif>
</host>

</vnuml>
```