Universitat de Girona

IMPROVING RESOURCE UTILIZATION IN CARRIER ETHERNET TECHNOLOGIES

Luis Fernando CARO PEREZ

ISBN: 978-84-693-1995-6
Dipòsit legal: GI-323-2010
http://www.tdx.cat/TDX-0303110-183013

Improving resource utilization in Carrier Ethernet technologies

Luis Fernando Caro Perez

Advisor: PhD. José Marzo

Doctorate Program in *Tecnologías de la Información*
Doctoral Thesis

Broadband Communications and Distributed Systems group

Thesis presented in fulfillment of the requirements for
the degree of PhD in Computer Engineering

Universitat de Girona

Girona, Catalonia, Spain

October 2009

Acknowledgments

I would like to thank my Thesis advisor PhD. Marzo for his complementary work in all aspects of this thesis. Thanks to Dimitri Papadimitriou for introducing me to Carrier Ethernet technologies and the topics of this research.

Thanks to the Department of Universities, Research and Information Society (DURSI) of the Government of Catalonia and the European Social Funds for the given grants. Thanks for the COST project action number 293: Graphs & Algorithms in Communication Networks for their financial and scientific support.

Thanks to PhD. Fernando Solano for all his time patience and friendship. Both being his friend and collaborating with him helped me learn most of what i needed to perform this research. Thanks to all of my friends both abroad and in Girona who supported me during these years. Thanks to my family who has always been there for me.

Abstract

Due to recent advances, Ethernet is starting to move from Local area networks to carrier networks. Nevertheless as the requirements of carrier networks are more demanding, the technology needs to be enhanced. Schemes designed for improving Ethernet to match carrier requirements can be categorized into two classes. The first class improves Ethernet control components only, and the second class improves both Ethernet control and forwarding components.

The first class relies only on improving Ethernet control components such as Multiple Spanning Tree Protocol (MSTP) and Rapid Spanning Tree Protocol (RSTP). With MSTP, several spanning trees can be created in the same Ethernet network, allowing to route traffic through different paths between a pair of nodes in the network. These technologies use this property to perform Traffic Engineering as well as to support protection by reserving resources to be used in case of network failure.

The second class relies on improving both Ethernet control and forwarding components. These techniques change the Ethernet forwarding plane by implementing forwarding based on an identifier, referred to as a label, defined by a subset of the Ethernet Medium Access Control frame header fields. Packets are sent through specific sequences of nodes denominated Label Switched Paths (LSP), giving similar functionality to that of Multi-Protocol Label Switching (MPLS). For this purpose each packet is marked with a label identifying the LSP through which it is sent. Each node uses the label as the index to look up in the forwarding table both the node where the packet needs to be forwarded to and the new label used to identify the packet in the next node. Two technologies under this class are considered in this document: Ethernet VLAN-Label Switching (ELS) and Provider Backbone Bridges - Traffic Engineering (PBB-TE).

Both ELS and PBB-TE use a different label size and scope than previous label based technologies such as MPLS (20 bits allowing up to 1048576 LSP per interface), in addition to not allowing to stack labels. For this reason, this thesis analyzes and compares label space usage for both architectures to ensure their scalability. The applicability of existing techniques and studies that can be used to overcome or reduce label scalability issues is evaluated for both ELS and PBB-TE. For ELS, a new routing algorithm to improve ELS label space usage is proposed. For PBB-TE, the label reutilization technique is formalized.

Additionally, none of the previous studies on label space usage in any of the existing label based forwarding architectures (e.g. MPLS) analyzes the impact of the topology characteristics on label space usage. Consequently, this thesis studies how topology characteristics affect the different label scopes. Both the number of states and the number of labels needed (relevant for label exhaustion)

considering label per link (used by ELS and MPLS) and destination scopes (used by PBB-TE) are analyzed.

Finally, despite the large number of studies that have been performed for the class of approaches improving Spanning Tree Protocols, they are always compared either among themselves or against the use of basic native Ethernet protocols. Additionally there is not any study that can determine when label based forwarding technologies have to be used instead of STP based approaches. Therefore, this thesis proposes an ILP to calculate optimal performance of this class of approaches and compares them with label based forwarding technologies to be able to determine, given a specific scenario, which approach to use.

Contents

*** Chapters containing contributions of the thesis are outlined by adding a superscript asterisk (*) at the end of their titles.**

List of Figures

List of Tables

Chapter 1

Introduction

This chapter gives an overview of the problem addressed and the complete thesis as well as the motivation and the desired objectives for this research work. Finally, the structure and contents of the rest of this document are outlined.

1.1 Motivation

In recent years, there has been an increasing demand for bandwidth combined with an exponential growth in the number of clients and network applications that require a carrier infrastructure. Such changes are placing a demand on carrier networks to constantly improve their bandwidth allocation flexibility and provisioning capability. Network providers are considering Ethernet as the inter-connection technology of choice for the metro (and even core) space. Ethernet allows more bandwidth to be offered per link by reducing capital expenditures (CAPEX) together with high-speed interfaces that range from 10Mb/s to 10Gb/s (100Gb/s being standardized at IEEE 802.3).

Nevertheless, as Ethernet is a LAN technology, native bridged Ethernet does not provide all the characteristics of a technology designed for carrier transport networks. Therefore the technology has to be improved in terms of scalability (for ensuring wide-scale deployment) and traffic engineering (for ensuring efficient network resource usage and resiliency) allowing its deployment in carrier networks. Efforts with the objective to extend native bridged Ethernet to fulfill these requirements can be classified in two classes.

The first class (which are know as "STP-based" technologies throughout the rest of this thesis) rely only on improving Ethernet control components such as Multiple Spanning Tree Protocol (MSTP) [802b] and Rapid Spanning Tree Protocol (RSTP) [80204]. With MSTP, several spanning trees can be created in the same Ethernet network, allowing the routing of traffic along different paths between a pair of nodes in the network. The STP-based technologies use this property to perform Traffic Engineering as well as to support protection by reserving resources to be used in case of network failure.

The second class relies on improving both Ethernet control and forwarding components. These techniques change the Ethernet forwarding plane by implementing forwarding based on an identifier, referred to as a label, defined by a subset of the Ethernet Medium Access Control (MAC) frame header fields. In

the case switching is performed on the service-VLAN Identifier (S-VID) value of the Tag Control Information (TCI) header field exclusively (defining a link-local label), the technique, known as Ethernet VLAN-Label Switching (ELS) [PDV05] allows the creation of Ethernet Label Switched Paths (LSP) giving similar functionality to that of Multi-Protocol Label Switching (MPLS). In the case switching is performed on the S-VID and the destination MAC address (defining a multi-component domain-wide label), the technique, known as Provider Backbone Bridges - Traffic Engineering (PBB-TE) [802a] encodes an end-to-end connection identifier on the forwarding plane. Both techniques can be combined with a distributed control plane such as Generalized MPLS (GMPLS) providing a set of specific extensions [Pea, Fea].

When comparing the two classes, STP-based technologies are characterized by being easier to implement given that only control components must be replaced from an regular Ethernet network whereas for implementing label based forwarding technologies (i.e. ELS and PBB-TE) almost all the equipment must be updated. Additionally STP-based technologies support only a subset of the carrier requirements making them unsuitable for providing certain carrier services which label based forwarding technologies support completely.

Finally, given that STP based approaches rely on spanning tree protocol to perform forwarding, routing can be limited given that traffic has to be routed using a limited number of trees. In the case of label based forwarding technologies, traffic is routed through label switched paths and routing is limited by the label size and scope of the specific technology. In ELS, a maximum of 4096 (2^{12} given by the 12 bit of the S-VID field) LSPs per link, can be forwarded. In PBB-TE a maximum of 4096 (2^{12}) LSPs per destination MAC address can be created.

Both ELS and PBB-TE use a different label size and scope than previous label based technologies. One of the most successful label based technologies, Multi Protocol Label Switching (MPLS), uses the same scope as ELS, but it has a longer label (20 bits), thus allowing up to 1048576 LSP per interface without considering stacking (not supported by ELS). Given that ELS labels have a significantly smaller size and intermediate nodes, i.e. E-LSRs are not capable of label stacking, it is possible that label space on certain links may have been exhausted before the full capacity of that link has been provisioned. In other words, the label size limitation could represent a new routing constraint, in addition to link capacity. To illustrate this constraint, let us consider the following example. In a carrier network, with an average link capacity of 10Gb/s, it could be said that the acceptable minimum bandwidth for each bandwidth request is equal to or higher than 1Mb/s given that traffic is being aggregated. In this network, given that the minimum bandwidth is 1Mb/s, the maximum number of LSPs that could traverse a link is 10,240. This example illustrates how the ELS label size could become a routing limitation (as $10,240 > 4,096$), while for MPLS it is not (as $10,240 < 1,048,576$). For PBB-TE is similar case, given that the technology uses a new size and scope (using a label considering MAC address of the destination), it is possible that label space on certain destinations may have been exhausted before the full capacity of the network has been provisioned. Therefore, the label size limitation could also represent a new routing constraint. For this reason, there is a need to analyze and compare label space usage for both ELS and PBB-TE to ensure their scalability.

Additionally, none of the previous studies on label space usage in any of the

existing label based forwarding architectures (e.g. MPLS) analyzes the impact of the topology characteristics on label space usage. Consequently, it is necessary to study how topology characteristics affect the different label scopes.

On the other hand, despite all the studies that have been performed for improving the scalability of routing in STP based approaches, they are always compared either among themselves or against the use of basic native Ethernet protocols without considering label based technologies. When deciding for implementing STP based or label based technologies, in an scenario where the network does not require the services not provided by STP based approaches, there is a need to determine if optimal performance of STP based approaches will be limited (because of the number of trees) when compared against label based technologies.

1.2 Objectives

The main objective of this thesis is to analyze and study the different carrier Ethernet technologies scalability and to compare and improve their performance. This is accomplished in three main tasks:

- To study and improve label scalability of ELS and PBB-TE. This includes to independently study each technology to determine if they present scalability issues and to finally compare their performance in terms of label space usage. This contribution is addressed in chapter 4 of this thesis.
- To study the influence of the topology characteristics on label space usage. This includes analyzing how topology characteristics affect the number of states and the number of labels needed (relevant for label exhaustion), considering the label scopes and techniques to improve label space usage available for carrier Ethernet technologies. This contribution is addressed in chapter 5 of this thesis.
- To study the routing performance and scalability of STP based approaches, proposing optimization models to calculate their optimal routing performance. Additionally, comparing them with label based forwarding technologies to determine, given a specific scenario, which approach to use. This contribution is addressed in chapter 6 of this thesis.

1.3 Contents

This document is organized into six Chapters including this one, plus the bibliography. Chapters 2 and 3 introduce the topics and concepts of the thesis. Chapters 4, 5 and 6, respectively present a main contribution of this thesis. They are organized as follows.

Chapter 2. This chapter introduces the fundamentals of the technologies considered in this document. The chapter introduces the basic principles of the Ethernet technology, including the spanning tree and VLAN protocols. It also introduces the concept of carrier Ethernet technologies and their classification. It explains in detail the frame formats and forwarding mechanism of Ethernet VLAN-Label Switching (ELS) and Provider backbone bridges - Traffic Engineering (PBB-TE).

Chapter 3. This chapter describes all the related work relevant to the objectives of this thesis. The chapter is divided in two parts; Part one describing the related work in label space usage on different label based forwarding technologies. It describes basic label space usage concepts as well as the existing techniques to improve label space usage. It summarizes studies performed on how to apply these techniques to optimize label space usage for MPLS and AOLS. The second part describes the related work that defines the different STP based technologies implementations and functionality. It also introduces the generalized routing problem in STP based technologies.

Chapter 4. This chapter focuses on studying and improving label scalability of PBB-TE and ELS. The applicability of existing techniques and studies (explained in Chapter 3) that can be used to overcome or reduce label scalability issues is evaluated for both architectures. The main contributions in this chapter include a new routing algorithm proposed to improve ELS label space usage [CPM08c, CPM09a] and the formalization of the label reutilization technique for PBB-TE [CPM08a].

Chapter 5. This chapter studies the influence of the topology characteristics on label space usage. It analyzes both the number of states and the number of labels needed (relevant for label exhaustion) considering labels per link (used by ELS and MPLS) and destination scopes (used by PBB-TE). The main contributions in this chapter include an analysis of the effect the topology characteristics incur on the improvement gained by applying available techniques to improve label space usage [CPM].

Chapter 6. This chapter evaluates the optimal performance of the STP based technologies and a comparison with label-based forwarding technologies is also presented. It analyzes both offline and online routing scenarios with and without protection. The main contributions in this chapter include an Integer Linear Program (ILP) that given a traffic matrix, calculates an optimal routing solution for the offline routing scenario with or without protection [CPM08b, CPM09b]. The performance of the schemes is compared and evaluated.

Chapter 7. The last chapter summarizes the most significant results of this work and outlines possible directions for future research.

Chapter 2

Carrier Ethernet Fundamentals

This chapter gives a general idea of Ethernet evolution from a Local Area Network (LAN) technology toward a carrier class technology for transport/aggregation networks. Both Ethernet technology principles and the advantages that have made Ethernet to be considered as the layer two technology of choice for carrier networks are explained. The limitations of native Ethernet given by its design toward Local Area Networks (LAN) are also introduced.

Finally, the technologies that enhance Ethernet forwarding and/or control components to overcome those limitations and fulfill carrier requirements are described. These technologies, defined as carrier Ethernet, include Ethernet VLAN-Label Switching (ELS) and Provider backbone bridges - Traffic Engineering (PBB-TE).

2.1 Ethernet Technology principles

Ethernet was designed as a frame based technology for Local Area Networks(LANs) and it has been standardized as IEEE 802.3. It defines a series of standards for the physical and data link network layers.

The technology was designed by Dr. Robert Metcalfe and David Boggs in the period of 1973-1975 [MB76], in 1985 it became a standard, namely the IEE 802.3. Ethernet has since become the most commonly used LAN technology worldwide. More than 85% of LANs are Ethernet based according to the International Data Corporation (IDC, 2000). The technology has transmission rates 10Mb/s 100Mb/s (Fast Ethernet) 1Gb/s (Gigabit Ethernet) and 10 Gb/s (10 Gigabit Ethernet). Large transmission speeds, simplicity and low prices are the key factors of its dominance in the LAN market.

Ethernet was initially based on the idea of computers communicating over a shared coaxial cable acting as a broadcast transmission medium. Nowadays it has evolved into a network where workstations are connected to switches via point to point links over either twisted pair or optical fiber cables. Workstations send data packets to the switches which in turn forward them to their destination. Each station is given a single 48-bit MAC address, which is used both to specify the destination and the source of each data frame.

Frames are the data packets format on the wire, even though the technology has changed considerably over the years, all Ethernet generations share the same frame formats and are compatible. Figure 2.1 shows the most common Ethernet Frame format, type II omitting the frame preamble. The Ethtype field is used to indicate which protocol (e.g. 802.1Q,802.1ad) is being transported in the frame.

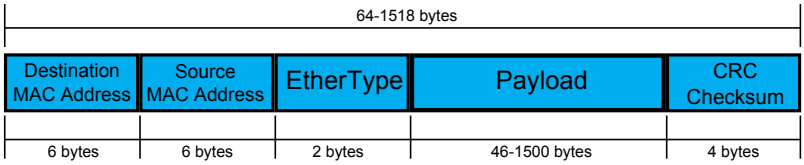


Figure 2.1: Type II Ethernet Frame format

Ethernet bridges learn which hosts are reachable from which ports by using the source MAC address of incoming packets. Each switch has a Filtering Database (FDB) where the address and port from which it came are stored. Then selectively copy frames from port to port comparing the frame destination MAC addresses with the FDB. When the destination MAC address is not registered in the FDB, the switch copies the frame to all ports.

2.1.1 The Spanning Tree Protocol (STP)

Ethernet switched networks can suffer from loops issues when there are several paths between two nodes. In order to avoid and prevent loops the Spanning Tree Protocol is used. The STP has been standardized as IEEE 802.1D [80203a] and is based on the Spanning Tree algorithm proposed by Radia Perlman [Per00].

STP allows switches to dynamically discover a subset of the topology that is loop-free (a tree) and where a path exists between any pair of network elements (spanning tree). Once the tree is discovered, all the ports that do not belong to the tree are blocked (frames received on blocked ports are dropped). Switches are constantly communicating to each other in order to keep track of network changes and activate or disable ports as required.

Spanning tree protocol information is carried in bridge protocol data units (BPDUs) which are a special type of frame. These BPDUs are exchanged regularly, every 2 seconds by default. When a link fails, a switch using the STP can take up to 50 seconds to activate the necessary but previously blocked ports. To reduce this the delay, Rapid STP (RSTP) was developed and standardized by IEEE 802.1w [80204]. RSTP can take up to 15 or 30 seconds in the worst case scenario.

2.1.2 Virtual LANs (VLANs)

A VLAN allows the creation of independent logical networks within a physical network. VLANs are assigned to set up logical segments of a LAN (like company departments) that should not exchange data using a LAN (they still can exchange data by routing). The protocol that defines how VLANs can be implemented and encoded on ethernet packets is IEEE 802.1Q [80203b]. As stated in the standard, VLANs offer the following benefits:

- VLANs facilitate easy administration of logical groups of stations that can communicate as if they were on the same LAN. They also facilitate easier administration of moves, additions, and changes in members of these groups.
- Traffic between VLANs is restricted. Bridges forward unicast, multicast, and broadcast traffic only to LANs that serve the involved VLAN.
- As far as possible, VLANs maintain compatibility with existing bridges and end stations.

VLANs are implemented by setting the EtherType value of the Ethernet header to Tag Protocol ID (TPID=hex 8100), identifying this frame as an 802.1Q frame. Two-bytes of Tag Control Information (TCI) are added after the TPID, followed by another two bytes containing the frame's original EtherType. Together the TPID and TCI bytes are called the VLAN Tag. A 802.1Q frame is presented in Figure 2.2.

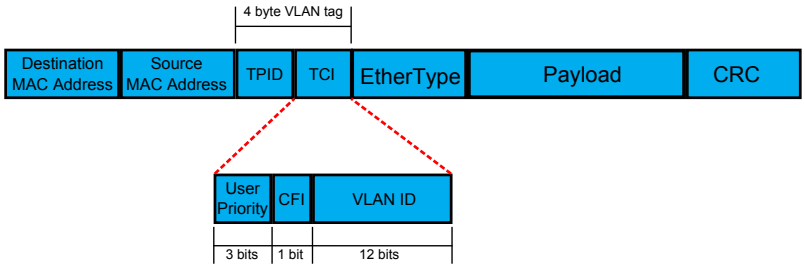


Figure 2.2: 802.1Q Frame

The TCI contains a 3-bit field storing the priority level for the frame. The use of this field is defined in IEEE 802.1p. It also contains a Canonical format indicator (CFI) 1 bit field which is used for compatibility with Token Ring. Finally, there is a 12 bit VLAN ID field which identifies the VLAN to which the frame belongs to.

The Multiple Spanning Tree Protocol (MSTP), originally defined in IEEE 802.1s and later merged into IEEE 802.1Q, defines an extension to the RSTP protocol to further develop the usefulness of VLANs. This "Per-VLAN" Multiple Spanning Tree Protocol configures a separate Spanning Tree for each VLAN group and blocks the links that are redundant within each Spanning Tree.

2.2 Introduction to Carrier Ethernet

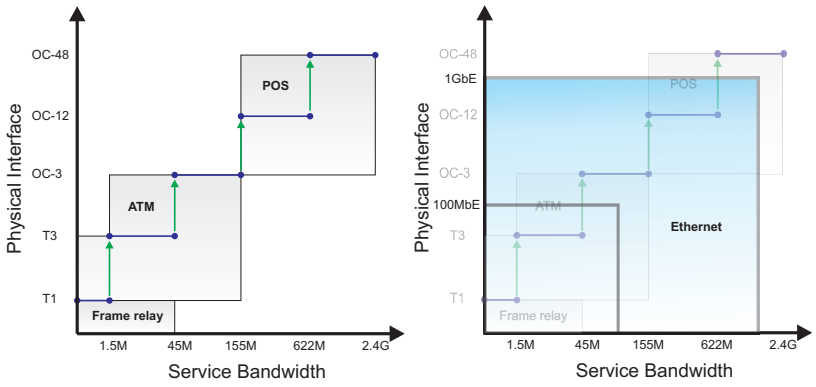
In the past couple of years, there has been an increasing demand for bandwidth combined with an exponential growth in the number of clients and network applications that require a carrier infrastructure. Such changes are placing a demand on carrier networks to constantly improve their bandwidth allocation flexibility and provisioning capability.

Simultaneously, Ethernet has been increasingly attracting service providers and the telecommunication community as the transport technology for Carrier networks.

2.2.1 Native Ethernet Advantages

The advantages of Ethernet include its high-speed interfaces that go from 10Mb/s to 10Gb/s (100Gb/s scheduled to be standardized by 2010) together with capital expenditure (CAPEX) reduction as Ethernet interfaces are cheaper due to their broad usage in all networking products.

Another characteristic is the technology flexibility, Figure 2.3a[1] illustrates the step function that occurs for TDM interfaces and non-Ethernet Layer 2 services as one increases bandwidth. The vertical axis indicates how the physical TDM interface changes as bandwidth increases. This requires replacement in equipment or interfaces cards as bandwidth needs cross bandwidth thresholds determined by the TDM digital hierarchy. Figure 2 illustrates an Ethernet service using a 100Mbps and 1Gbps Ethernet interface. In both cases, the same Ethernet protocol is used and hence, as bandwidth needs cross the 100Mbps threshold, a new interface card may be needed. Most interfaces today support 10Mbps, 100Mbps and 1Gbps over the same interface card so there would be no need for a new interface.



(a) TDM interface (b) Ethernet interface

Figure 2.3: Service Bandwidth per interface

Additionally, edge and even core routers are progressively populated with Gigabit Ethernet interfaces, allowing the interconnection of routers by an Ethernet aggregation network.

2.2.2 Native Ethernet Limitations

Regardless of the advantages, when Ethernet is positioned as a transport/aggregation technology (e.g. for metro networks), the technology inherits the properties of its design, which is oriented toward facilitating the interconnection of various LAN segments by a reduced set of bridges.

Ethernet is a connectionless broadcast-access technology that relies on the Spanning Tree Protocol (STP) and its enhanced versions, such as the rapid spanning tree protocol (RSTP), to create loop free topologies. The bridged

[1] Figure taken from [San03]

Ethernet properties were specifically designed for LAN and other access environments. However, the carrier aggregation networks, where Ethernet is progressively extending, have properties that are not comparable to networks where Ethernet traditionally applies. Such environments, when considering Ethernet as a candidate technology, should address:

- Ethernet Media Access Control (MAC) address space lookup: Ethernet MAC frame forwarding uses hash-based table lookup that limits MAC table size due to memory consumption and non-deterministic lookup time. Hence, carrier Ethernet frame forwarding should not be MAC address dependent. Moreover, Ethernet aggregation should ideally be independent of the number of interconnected clients and provide isolation of traffic from different users (with no limitations on the number of clients connected to the network).

- Ethernet MAC address learning: It relies on (routing by) flooding of unknown unicast MAC frames, which is appropriate for LAN environments but has several shortcomings when applied to meshed aggregation environments. Firstly, the flooding of unknown MAC frames across the spanning tree topology creates unnecessary processing overhead (aging, filtering, etc.). Secondly, bridges require Filtering Database (FDB) update during STP re-convergence thus leading to slow recovery.

- Dynamic, flexible and resource-efficient set up of Ethernet data paths: Another major limitation of the current control components for bridged Ethernet networks is its lack of traffic engineering capabilities. Due to the aggregation network size when compared to LAN networks, the number of blocking links determined by the Spanning Tree Protocol leads to inefficient use of network resources. The Multiple Spanning Tree Protocol (IEEE 802.1s) was added for basic traffic engineering in VLAN bridged Ethernet networks. The protocol allows the use of multiple spanning trees for traffic, belonging to different VLANs, to flow over different paths within the bridged Ethernet network. By using IEEE 802.1s, it is possible to define which VLANs should preferentially use certain links. However, this technique is static and complex to configure (in particular, for meshed environments), and still leads to an inefficient allocation of link resources. In other words, the usage of Multiple Spanning Tree Protocol for traffic engineering purposes is limited in native bridged Ethernet networks. Carrier Ethernet shall provide route computation and selection (based on various network and service constraints) during the provisioning of Ethernet data paths. Using this flexibility, providers can make use of traffic engineering techniques to optimize network resource usage through load sharing and route paths around bottlenecks to less loaded links (i.e. avoiding the hyper-aggregation problem).

- Network recovery: The Rapid/Spanning Tree Protocol (IEEE 802.1w/802.1d), being a Distance Vector protocol, has inherent limitations that make "fast recovery" time performance objectives difficult to accomplish. (R)STP is used to construct a loop-free logical tree topology, which is originated at the root bridge, with leaves and branches spanning all bridges of the entire Ethernet broadcast domain or sub-domain. The IEEE 802.1d STP

is based on a break-before-make paradigm. It takes up to 50 seconds to recover from a link failure. Subsequent attempts, such as the RSTP, to make it less conservative by considering a make-beforebreak approach with faster convergence time (in the range of two seconds) do not fundamentally solve the initial problem of slow convergence compared to expectations for carrier class network.

In summary, native bridged Ethernet does not properly address the scalability (for ensuring wide-scale deployment) and traffic engineering (for ensuring efficient network resource usage and resiliency) required by network providers. It should be noted that Virtual LANs (VLANs) tagging, defined in IEEE 802.1Q and its extension, do not change these observations.

2.2.3 Ethernet in carrier networks

Given a carrier network implementing Ethernet as the layer 2 technology, there are two main technology directions to provide the necessary carrier grade services as well as to fulfill network provider requirements.

The first direction is to use a data-carrying mechanism over Ethernet such as Multi Protocol Label Switching (MPLS) [RVC01], in an MPLS network, provider requirements can be fulfilled and carrier grade services can be supported. However MPLS possesses capabilities and mechanisms that are not relevant to transport networks operations and that do not provide support for critical transport functionality. Because of this reason, the IETF and ITU-T are working in collaboration in the design of a MPLS Transport Profile (MPLS-TP). MPLS-TP combines the necessary existing capabilities of MPLS (excluding the unnecessary ones) with additional minimal mechanisms so that it can be used in a transport network.

The second direction, which in this document is considered as Carrier Ethernet, refers to the technology resulting from the extensions of the native bridged Ethernet forwarding and/or control plane components to address the needs of transport/aggregation networks. Given that the main focus of this document is to study carrier Ethernet technologies, their design is further explained in section 2.3.

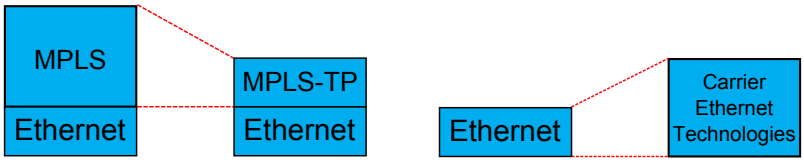


Figure 2.4: Protocol stacks

Figure 2.4 shows an illustration of the protocol stack of the two directions, both aim for the same level of capabilities and services. Nevertheless while MPLS-TP is a reduction of MPLS, carrier Ethernet is an enhancement of the native Ethernet layer.

2.3 Carrier Ethernet technologies

As already defined in the previous section, carrier Ethernet technologies are extensions of native bridge Ethernet developed to address the needs of transport/aggregation networks. Carrier Ethernet technologies can be classified into two classes.

The first class (which are denominated STPbased technologies through the rest of this document) rely only on improving Ethernet control components such as Multiple Spanning Tree Protocol (MSTP) and Rapid Spanning Tree Protocol (RSTP), without improving native Ethernet forwarding components. More details on this class are presented in section 3.2.

The second class of Carrier Ethernet technologies redefine both bridged Ethernet forwarding and control components enabling an Ethernet network to:

- Perform forwarding based on a transport label, not on a customer MAC address.
- Establishment of (logical) data paths, similar to MPLS LSP.
- Posses a centralized management or distributed control plane.

Two carrier Ethernet technologies belonging to this class, are studied in this document, Ethernet VLAN-Label Switching (ELS) and Provider Backbone Bridges - Traffic Engineering (PBB-TE).

2.3.1 Ethernet VLAN-Label Switching (ELS)

ELS[PDV05, ea07a] enables the creation of logical data paths established by using constraint-based routing mechanisms provided by a control plane such as Generalized Multi-Protocol Label Switching (GMPLS). The idea behind this approach is to prevent both the forwarding and the control plane from dealing with any Ethernet MAC address in order to maintain independence and transparency in the data plane addressing space.

IEEE 802.1ad frame

ELS uses the Ethernet frame described by the Provider Bridges (PB) standard, defined as IEEE 802.1ad [80205]. This standard is an extension of the IEEE 802.1Q, it intends to develop an architecture and bridge protocols to provide separate instances of the MAC services to multiple independent users of a Bridged Local Area Network in a manner that does not require cooperation among the users, and requires a minimum of cooperation between the users and the provider of the MAC service. An illustration of the IEEE 802.1ad Ethernet frame is presented in Figure 2.5.

IEEE 802.1ad allows the separation of the VLAN ID space by enabling an Ethernet frame to have two VLAN IDs instead of just one. The customer VLAN-ID (C-VID) that identifies VLANs under the administrative control of a single customer of a service provider and the service provider VLAN-ID (S-VID) TAG field that identifies VLANs used by a service provider to support different customers. This means that in a Provider Bridge Network (PBN), each customer and the service provider have their own VLAN space of 4096 VLANs (C-VID for the customer and S-VID for the provider) without the need for any cooperation among customers.

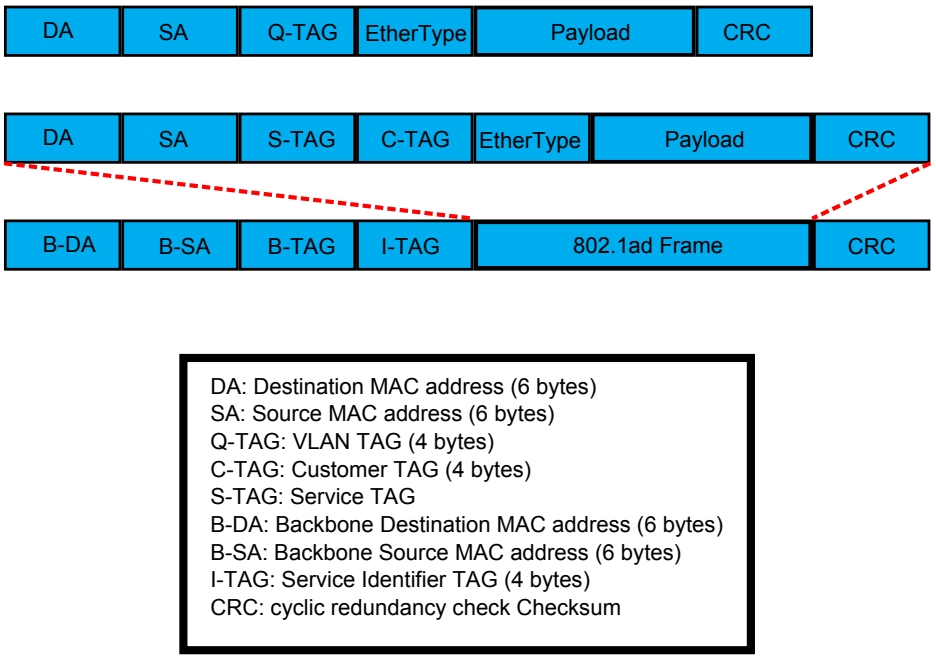


Figure 2.5: 802.1 frames

ELS forwarding mechanism

ELS performs label based forwarding by encoding the label in the service provider VLAN-ID (S-VID) TAG field of the IEEE 802.1ad frame. These labels, referred to as S-VID labels, are assigned and interpreted locally. The Ethernet S-VID label space has a link local scope and significance, thus providing for 4096 (2^{12}) values per interface. Using this label semantic, Ethernet MAC frame switching based on the S-VID label is performed at any device interface able to process this information field.

Thus, ELS enhances the Ethernet MAC frame with the properties of a label switching technology by providing a label semantic to its header. This feature is defined without modifying the IEEE 802.3 frame header format (ensuring interoperability with legacy Ethernet switches). The implication is that ELS does not rely on Ethernet MAC address learning (classical Ethernet switches execute this learning process by flooding unknown unicast Ethernet MAC frames) and MAC destination address (DA)-based forwarding.

The logical data paths established using ELS are denoted Ethernet label switched paths (LSP). Figure 2.6 describes the label operations along an Ethernet LSP. Intermediate nodes are denoted Ethernet label switching routers (E-LSR). The functionality of E-LSRs where the LSP starts and ends is referred to as Ethernet Label Edge router (E-LER). When a native Ethernet frame arrives at the ingress LSR, its E-LER function based on the information of the frame header pushes the correct label (i.e. by adding an S-TAG with the appropriate S-VID value). Then, the Ethernet VLAN-labeled frame is forwarded along the Ethernet LSP. At each E-LSR, the label is swapped (i.e. the incoming S-VID is translated into an outgoing S-VID as defined in IEEE 802.1ad). When the

frame reaches the egress LSR, its E-LER function pops the label (removing the S-TAG and therefore the S-VID). Finally, the frame is sent as a native Ethernet frame to its destination. E-LSRs are capable of performing swap operations only on labeled frames. The nodes capable of performing label push and pop are E-LSRs with LER functionality. The ELS control plane relies on the unified traffic engineering capabilities of GMPLS extended by [PDV05].

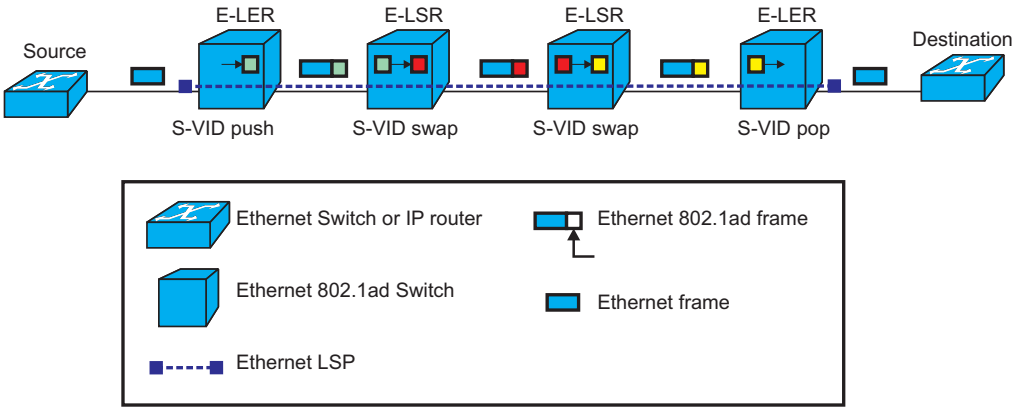


Figure 2.6: ELS label operations

Given the scope and encoding of ELS labels, in an ELS network, a maximum number of 4096 (2^{12}) LSP can be established traversing the same interface.

2.3.2 Provider Backbone Bridges - Traffic Engineering (PBB-TE)

PBB-TE is under definition at IEEE in the context of the 802.1Qay [802a] effort. It also enables an Ethernet network to create logical paths by using constraint-based source based routing. However, the label encoding used is different from ELS.

IEEE 802.1ah frame

PBB-TE uses the Ethernet frame described by the Provider Backbone Bridges (PBB) standard, defined as IEEE 802.1ah [80208]. The standard is an extension of the IEEE 802.1ad standard. In addition to VLAN space separation, PBB adds Ethernet MAC address space separation (between client and network) as it enables to encapsulate a client Ethernet frame (using client MAC address space) into a network Ethernet frame (using network MAC address space).

In PBB, Backbone Edge Bridges (BEB): i) encapsulate and de-encapsulate incoming (service) frames within backbone MAC frames and ii) insert encapsulated service frames and forwarding encapsulated service frames over the PBB network (PBBN). Within the PBBN, Backbone Core Bridges (BCB) forward the encapsulated frames. A PBBN is illustrated in Figure 2.7.

An illustration of the IEEE 802.1ah Ethernet frame is presented in Figure 2.5. The Backbone MAC frames used to encapsulate service frames include Backbone MAC Destination Address (Destination B-MAC or B-DA), the Backbone MAC Source Address (Source B-MAC or BSA), the B-TAG (12 bit B-

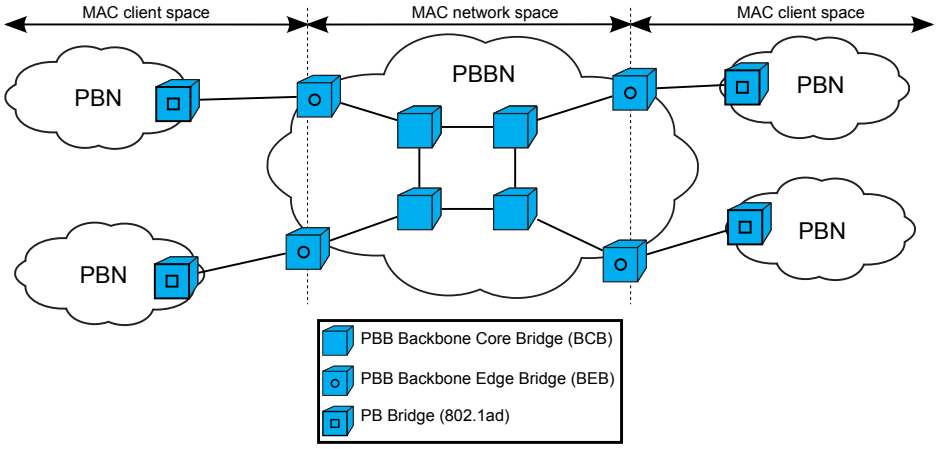


Figure 2.7: A PBB network

VID), the I-TAG (incl. 24 bit ISID), and the Client Ethernet MAC frame. The I-TAG allows the carrier to assign QoS parameters and define a unique customer identifier (I-SID).

PBB-TE forwarding mechanism

PBB-TE enables a PBB network to create logical paths by using constraint-based source routing. As in ELS, the logical paths established using PBB-TE are also called Ethernet Label Switched Paths (LSP). PBB-TE nodes can create Ethernet LSPs and forward frames based on a combination of the backbone VLAN id (B-VID) and backbone destination MAC address (B-DA) fields. Operation performed at intermediate Ethernet switches is equivalent to a label switching (not swapping) operation. Using this equivalence, the scope of the label is domain wide, meaning that the label is globally unique and end-to-end significant. Figure 2.8 gives an example of logical paths created using PBB-TE. In the example there are 4 logical paths established, two from PBB1 to PBB2 and two from PBB1 to PBB3. The nodes forward the frames based on the B-VID and B-DA fields, therefore two logical paths with different destinations can have the same B-VID value.

As said before, the rationale for PBB-TE is to support connection-oriented traffic engineered point-to-point trunks in a PBB network established using a provisioning system. Some B-VIDs are reserved for PBB-TE and used to identify the PBB-TE data paths. Each PBB-TE data path is identified from an ingress PBB node by <(B-SA), B-DA, B-VID>. Frames are encapsulated in the same way as any PBB traffic and forwarded based on <B-DA, B-VID>. So, forwarding hardware must perform a 60-bit lookup (B-VID (12-bit) + B-DA (48-bit)) to forward Ethernet MAC frames in the PBBN.

For compatibility reasons, PBB-TE preserves global uniqueness and semantics of MAC addresses as interface but redefines semantics associated to a subset of B-VID values (from the behavior defined in IEEE 802.1ah). In this subset, the B-VID value space is only significant when combined with a destination B-MAC address. Hence, the B-VID space can be considered as an individual instance identifier for one of a maximum of 4096 point-to-point or multipoint-to-point

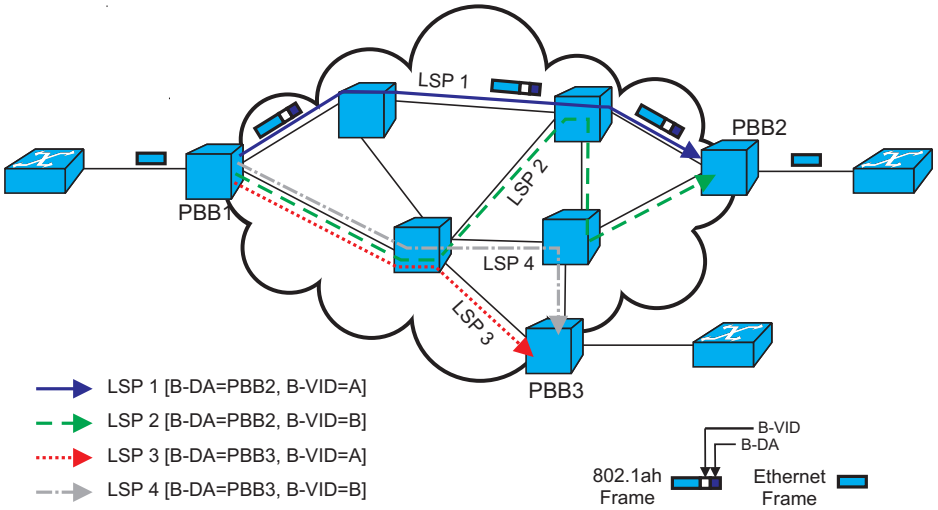


Figure 2.8: Provider backbone bridges - Traffic Engineering example

data paths. In this subset, B-VID value space is not unique on an Ethernet sub-network basis, though the <B-DA, B-VID> tuple is unique. This choice results in a single unique and invariant identifier (or label) associated with the path termination and not a sequence of local identifiers associated with the individual link terminations. PBB-TE introduces thus into the Ethernet data plane a connection identification functionality associated to the concatenated (B-SA +) B-DA + B-VID field (108 bits). In other terms, the B-DA and B-VID fields define a composed "label" whose value space is domain-wide. Due to the fact that the B-DA part of the label is not assigned but given, we define PBB-TE label scope as a per destination scope. Due to its label forwarding mechanism, any service or functionality relying on label swapping (e.g. segment protection) is not supported by PBB-TE.

Given the scope and encoding of PBB-TE labels, in a PBB-TE network, a maximum number of 4096 (2^{12}) LSP can be established ending at the same destination.

2.4 Chapter remarks

In this chapter the fundamentals of carrier Ethernet are discussed. A map of the technologies introduced is illustrated in Figure 2.9. The scope of this document lies within the defined carrier Ethernet technologies.

In this chapter the details of the two technologies considered in this document that improve control and forwarding components are explained. Each of their label scopes used in order to perform label-based forwarding was described. ELS uses a link local scope, which together with the label encoding limits the technology to have 4096 (2^{12}) LSP using the same interface. PBB-TE on the other hand uses a per destination scope, which together with the label encoding limits the technology to have a maximum of 4096 (2^{12}) LSPs ending at the same destination. Both ELS and PBB-TE use a different label size and scope than previous label based technologies such as MPLS (20 bits allowing up to 10240

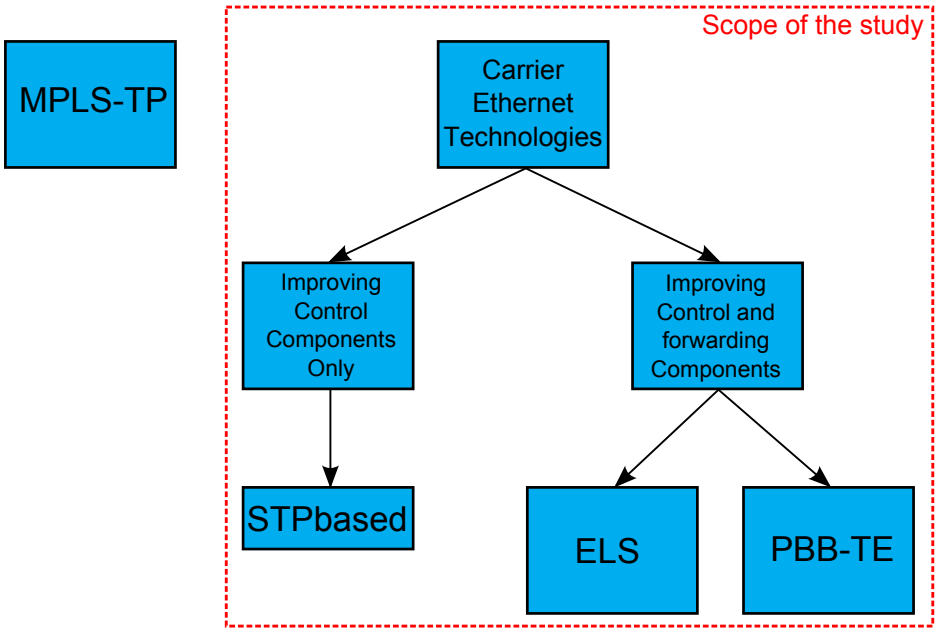


Figure 2.9: Technological map

LSP per interface), in addition to not allowing to label stacking. For this reason, one of the main contributions of this thesis is to study and compare label space usage for both architectures.

Additionally in this chapter the concept of the STPbased carrier Ethernet technologies is introduced. STPbased technologies enhance only Ethernet control components conserving native Ethernet forwarding paradigms. Another main contribution of this thesis is to evaluate the optimal performance of the STPbased technologies and compare it against label-based ones such as ELS and PBB-TE.

Chapter 3

Related Work

In this chapter, a brief summary of the related work performed in the areas of this thesis is presented. Previous studies on how to improve label space usage on different label-based architectures are explained. Both the metrics used to measure the label space usage and the available techniques to improve it are introduced.

Following this, the studies that propose and design the STPbased technologies are explained. Finally, a general summary that classifies the proposals based on their supported characteristics is presented.

3.1 Label space usage studies

To the best of our knowledge, there have not been any label space studies for carrier Ethernet technologies. Nevertheless, label space reduction has been studied for other label-based architectures like MPLS and All Optical Label Swapping (AOLS). A description of these studies is given in this section.

3.1.1 Label space usage basic concepts

Before explaining the related work on label space usage, some basic concepts about label space usage in label-based forwarding architectures need to be detailed.

Label scope

The scope of an architecture label space, or label scope, refers to the domain of significance of the labels (from that space) in which they can be assigned to the paths. All the label-based architectures studied or referenced in this document use either per link or per destination scopes.

When labels have per link scope, each node can change the value of the packet label when forwarding it (label swapping). Therefore label assignment is performed on a per link basis. Each LSP on each link has an assigned label that is different from the labels of the other LSPs traversing the same link. In this architecture, the number of bits of the field in which the label is encoded determines the maximum number of LSPs that can traverse a link. Architectures that use labels per link scope include MPLS, AOLS and ELS.

When labels have per destination scope, nodes forward packets based on their label, comprising of an identifier uniquely assigned by the destination node of the LSP. If this identifier is an address associated to the destination node and/or interface, a demultiplexing identifier can be included in the label definition, allowing each destination to terminate more than one distinct LSP per node/interface. The value of the label of each packet remains constant through all the links of the LSP (nodes do not perform label swapping; they only make their forwarding decision based on the label value). This means that label assignment is performed on a per LSP basis. Each LSP has one assigned label that is different from the labels of the other LSPs ending at the same destination. In this architecture, the number of bits of the field encoding the label demultiplexing identifier determines the maximum number of distinct LSPs that can terminate at a given destination. Architectures that use labels per destination scope include PBB-TE.

Label space usage metrics

Label space usage is measured by means of two different metrics: the number of forwarding states per node and the number of labels used relative to the specific label scope. The number of forwarding states is equal to the number of [labels, outgoing interface] or [labels per destination address] couples needed to be identified at each node to enable correct packet forwarding. When techniques for improving label space usage are not used, the number of forwarding states is equal to the number of LSPs traversing the node. The number of used labels relative to the specific scope is equal to the number of different assigned labels in each scope (link or destination). When the techniques for improving label space usage are not used, the number of used labels is equal to the number of LSPs traversing a specific link (for labels per link scope) and to the number of labels ending at a specific destination (for labels per destination scope).

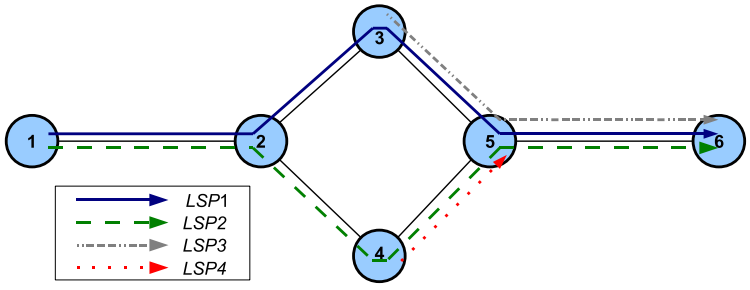


Figure 3.1: Example

To illustrate these metrics let us consider the example of Figure 3.1: Given four LSPs $LSP1 = (1, 2, 3, 5, 6)$, $LSP2 = (1, 2, 4, 5, 6)$, $LSP3 = (3, 5, 6)$ and $LSP4 = (4, 5)$, in a labels per link scenario, we could assign the following labels to each LSP: to $LSP1$ (label A in link $(1, 2)$, label B in $(2, 3)$, label C in $(3, 5)$, label A in $(5, 6)$), to $LSP2$ (label B in link $(1, 2)$, label A in link $(2, 4)$, label A in $(4, 5)$, label B in $(5, 6)$) to $LSP3$ (label A in $(3, 5)$, label C in $(5, 6)$) and to $LSP4$ (label B in $(4, 5)$). The assignment is illustrated in Figure 3.2. The number of forwarding states (F. states) in this case would be three states for

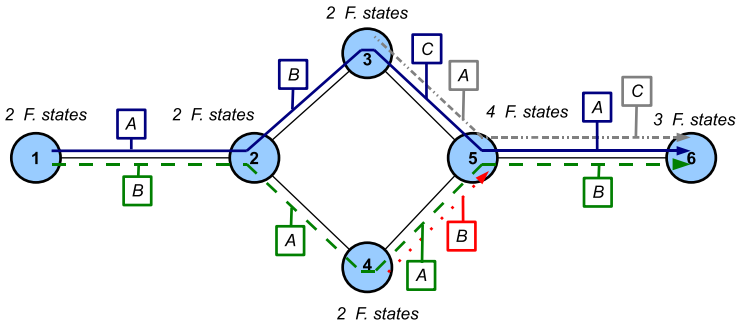


Figure 3.2: Labels per link assignment example

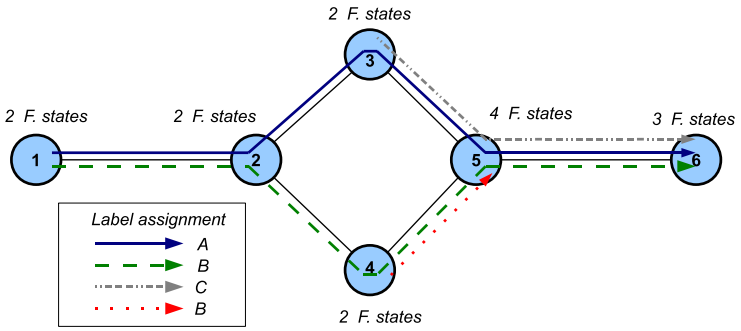


Figure 3.3: Labels per destination assignment example

node 6, two states for nodes 1, 2, 3 and 4, four states for node 5. On the other hand the number of used labels would be two for link $(1, 2)$, one for link $(2, 3)$, one for link $(2, 4)$, two for link $(3, 5)$, two for link $(4, 5)$ and three for link $(5, 6)$. In a labels per destination scenario, as described in Figure 3.3, we could assign the following labels to each LSP: label A to $LSP1$, label B to $LSP2$, label C to $LSP3$ and label B to $LSP4$. The number of forwarding states would be the same as in the labels per link scenario. However, the number of used labels would be three for node 6, and one for node 5, as node 6 and 5 are the only destinations.

3.1.2 Techniques for improving label space usage

Several techniques may be used on label-based architectures that assign to several LSPs the same label, thus improving label space usage.

Label merging

Label merging can be used in technologies for which the forwarding operation involves label swapping. Of the considered technologies, it can only be applied to the ones with labels per link scope. Label merging assigns the same label to two or more LSP in a continuous and common segment (a continuous sequence of links) that goes from any common outgoing link (of a common intermediate node) to the same destination. To be merged, all LSPs must follow the same

path from the intermediate node to the destination. Whether the two LSPs intersect at an intermediate node or not before intersecting at the common outgoing link is not a constraint. Label merging is able to reduce the number of labels used per link and the number of forwarding states per node.

In the example of Figure 3.1, in a labels per link scenario with label merging applied, labels could be assigned as follows: to $LSP1$, label A in link $(1, 2)$, label B in $(2, 3)$, label A in $(3, 5)$, label B in $(5, 6)$; to $LSP2$, label B in link $(1, 2)$, label A in link $(2, 4)$, label B in $(4, 5)$, label B in $(5, 6)$; to $LSP3$ label A in $(3, 5)$, label B in $(5, 6)$; and to $LSP4$, label A in $(4, 5)$. The number of forwarding states in this case would be two states for nodes 1, 2, 3, 4 and 5, and one state for node 6. The assignment is illustrated in Figure 3.4. Additionally, the number of used labels would be two for link $(1, 2)$, one for link $(2, 3)$, one for link $(2, 4)$, one for link $(3, 5)$, two for link $(4, 5)$, and one for link $(5, 6)$.

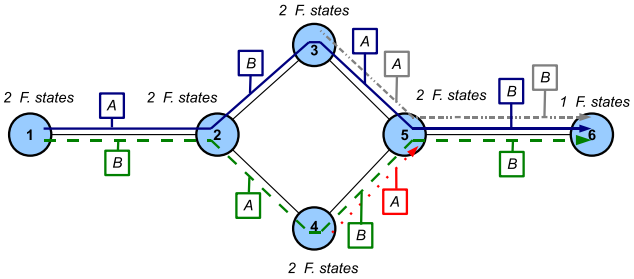


Figure 3.4: Label merging example

Inverse trees

Inverse trees are a modification of label merging for architectures that use a label per destination scope. Inverse trees allow the same label to be assigned to two or more LSPs ending at the same destination. They allow two or more LSPs to share a label if they intersect in only one common segment from an intermediate node to their destination. Inverse trees reduce the number of labels used per destination and the number of forwarding states per node.

In the example of Figure 3.1, in a labels per destination scenario when inverse trees are applied, the only LSPs that can share a label are $LSP1$ and $LSP3$. This is because they have only one common segment $(3, 5, 6)$ and it reaches their destination. $LSP1$ and $LSP2$ can not share a label because they have two common segments $(1, 2)$ and $(5, 6)$. The number of forwarding states in this case would be three states for node 5 and two states for the rest. The assignment is illustrated in Figure 3.5. Additionally, the number of used labels would be two for node 6, and one for node 5.

It is important to note that in several related works the inverse trees technique is called label merging. Nevertheless, in this document, due to the fact that inverse trees can save less number of labels than label merging, and part of this document's objective is to compare the two techniques, a different term is used specifically to differentiate the two.

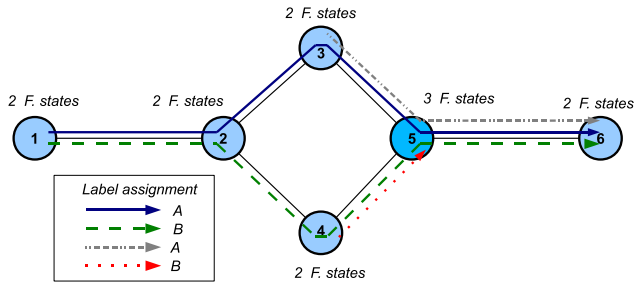


Figure 3.5: Inverse trees example

Asymmetric Tunneling (AT)

Asymmetric Tunneling is a technique for reducing the number of labels used, it can be used in any label switched network (e.g. MPLS) where nodes are capable of performing label stacking. The technique consists of pushing the same label in a set of LSP, that share a common segment, all LSRs of the segment regard the LSP as only one. The common segment must have at least 2 hops. An example of AT can be appreciated in Figure 3.6. In the example there are two LSP with a common segment (2-3-4-5). To perform AT, node 2 pushes label X to the two LSP and node 4 pops it, thus nodes 3 and 4 regard the two LSP as one. Even though the segments end at node 5 the label is popped on node 4, this is due to the Penultimate Hop Popping principle explained in [RVC01], and is the main reason why the segment must have at least 2 hops. Another characteristic of AT is that any node can push labels so any number of LSP can be added at intermediate nodes of the segment, but all the LSP belonging to the tunnel must have the label popped at the same node.

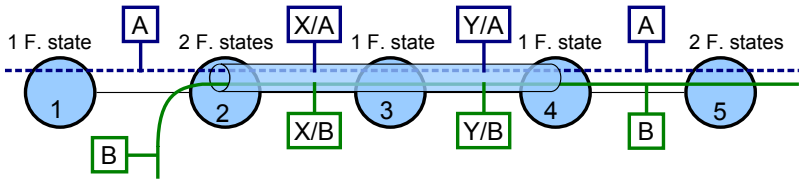


Figure 3.6: AT example

Asymmetric Merged Tunneling (AMT)

AMT is a technique that uses the combination of label stacking and label merging, it can be seen as a mixed version of label merging and AT, that preserves their advantages. It can be described as a merging of asymmetric tunnels into a single connection or as a way to merge LSPs with a common segment that does not end at their destination.

Formally, given 4 LSPs routes and 3 network segments (a segment is an ordered set of nodes):

- $S_1 = \{n_i, n_k, ...n_j\}$
- $S_2 = \{n_a, n_b, ...n_c\}$

- $S_3 = \{n_x, ...n_y, n_z\}$
- $LPS_1 = \{n_q, ...n_l, S_1, S_3, n_p, ...\}$
- $LPS_2 = \{n_h, ...n_m, S_1, S_3, n_w, ...\}$
- $LPS_3 = \{n_f, ...n_o, S_2, S_3, n_v, ...\}$
- $LPS_4 = \{n_d, ...n_r, S_2, S_3, n_t, ...\}$

Then an Asymmetric Merged Tunnel can be built by performing the following operations:

1. n_i pushes the same label to LSP_1 and LSP_2, lets denominate the label X.
2. n_a pushes the same label to LSP_3 and LSP_4, lets denominate the label Y.
3. n_x swaps both label X and label Y and assigns them the same label, lets denominate the label Z.
4. n_y pops label Z, so that n_z receives the packets with their original label and forwards them to the next node (i.e. n_p, n_w, n_v, n_t).

Figure 3.7 illustrates an example of an AMT. In the example $S_1 = \{2, 3\}$, $S_2 = \{1, 3\}$, $S_3 = \{3, 4, 5\}$.

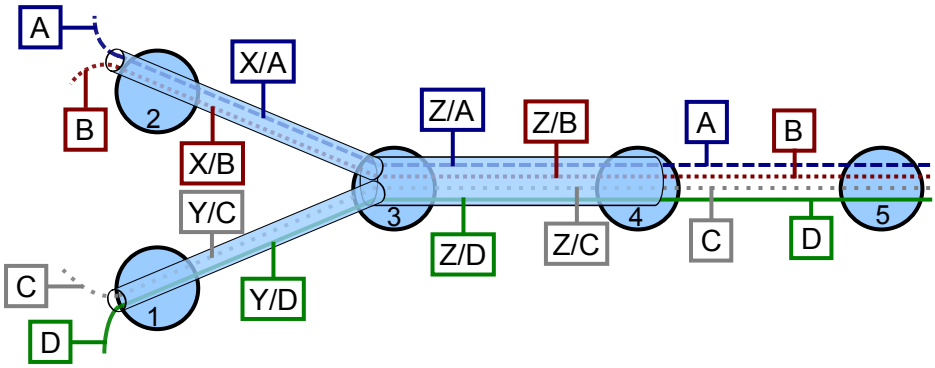


Figure 3.7: AMT example

3.1.3 Label space studies in MPLS

Even though a 20 bit label encoding does not represent a routing constraint, the main reasons why label space reduction has been considered in MPLS are the following:

- To offer MPLS-based Virtual Private Network (VPN) services to thousands of customers, ISPs will need to set up and handle thousands of MPLS LSPs for the VPN endpoints.
- Some protection mechanisms duplicate label space needs.
- Reducing label space simplifies network management and hence OPEX.

All the related work described in this section considers that all given LSPs have equivalent Forwarding Equivalence Classes (FEC) as recommended in the MPLS architecture [RVC01]; otherwise, LSPs cannot be joined (either merged or stacked).

Two label space reduction scenarios are considered in the literature, with and without re-routing.

The label space reduction with routing scenario is studied in [AT03]. For a network with N nodes and M edges, [AT03] presents an offline routing algorithm that uses routing table sizes of at most $(N + M)$ labels, where previous algorithms use at most (N x M) labels. The algorithm receives the network topology, the demand matrix and a given routing solution S as input. Then, the algorithm returns another routing solution S' such that for each link e in the network, the load of e in S' is not higher than S. If S' is optimal to a cost function, S' preserves the optimality. S' has the characteristic of having table sizes of at most $(N+M)$. The algorithm uses label merging (which means nodes are able to perform label swapping) to reduce the label size. It uses an Integer Linear Program (ILP) based on the multi-commodity flow model.

Related work that studies the label space reduction without routing scenario aims to optimize the number of labels used regardless of the LSP routes (i.e. the LSP routes are already given). In the label space reduction without re-routing scenario the problem is stated as follows: given a network and a set of LSP routes, the goal is to determine the operations performed at each node (swapping or stacking) so that the total number of labels used in the network is minimized. Previous work in this scenario can be classified based on the techniques studied:

- [SMY00] and [BGN03], study the problem of minimizing the number of used labels by using label merging. They state that the problem cannot be solved optimally with a polynomial algorithm (NP-complete), since it involves a hard decision problem. In these studies label merging is considered using a tree-shape consideration making the problem equivalent to using Inverse Trees instead of merging. Thus proving that the problem of minimizing the number of used labels using Inverse Trees is NP-complete. Additionally in [SFM08], it is shown that for MPLS (or any other architecture capable of label swapping), the tree-shape consideration can be overridden (considering label merging as described in this document) making it possible to perform label merging in polynomial time with guarantees that the optimal solution can always be found.

- To the best of our knowledge it has not been demonstrated that the problem of optimally minimizing the number of used labels using asymmetric tunneling or asymmetric merged tunneling is NP-complete. Nevertheless, several algorithms have been proposed in order to solve the problem, some of them include the Longest Segment First (proposed in [SFDM05a, SFDM05b]) and the Most Congested Space First Algorithm (proposed in [SFM05]).

- Two methods have been proposed to solve the problem of optimally minimizing the number of used labels using asymmetric merged tunnels. The first one is a Brute-Force model (B-F model) and the second one is the Decompose & Match framework. The B-F is an optimization model that

has as a objective function to minimize the total number of labels, it has the disadvantage that the model is complex and can take time to find optimal solutions. The Decompose & Match framework is a combination of an algorithm for performing pre-computations and an optimization model with the same objective that B-F, but more efficient and simpler.

3.1.4 Label space studies in AOLS

All Optical Label Swapping (AOLS) is an Optical Packet Switching (OPS) architecture. AOLS nodes are capable of reading the incoming label of a packet, replacing it with the proper outgoing label (label swapping) and performing wavelength conversion if necessary. The node performs all these operations without converting the packet to the electronic domain; in other words, the operation is completely optical. The AOLS architecture is described in [RKM+05].

The cost of deploying AOLS grows linearly with the number of labels that the network is able to support. Due to this fact, in AOLS, label space reduction represents a top priority over QoS traffic parameters. Therefore, new routing schemes designed to reduce the number of used labels instead of optimizing traffic engineering metrics are needed.

Considering this fact [ea08] studies the label space reduction problem together with the routing problem in AOLS. The objective is to reduce the total number of labels used in the network.

In [ea08] an extension of the AOLS architecture to perform label stacking is studied. The tradeoff of performing label merging and/or stacking are analyzed using an ILP model. Two algorithms are also proposed. The first being a routing algorithm based on CSPF which routes a traffic demand matrix aiming at choosing paths so they share the maximum number of links. The second algorithm establishes the label assignment of the routes in order to reduce or minimize the total number of labels. Both algorithms have been designed to take advantage of the properties of label stacking.

3.2 Carrier Ethernet STPbased technologies

There has been considerable work done in the study of STP based technologies. As mentioned in the previous chapter these approaches rely only on improving Ethernet control components such as Multiple Spanning Tree Protocol (MSTP) and Rapid Spanning Tree Protocol (RSTP), without improving native Ethernet forwarding components. They combine an external routing computation and decision process as well as re-configuration mechanisms so as to elevate the STP limitations.

As explained in Section 2.1, the MSTP protocol allows to have a specific spanning tree for routing packets of each VLAN of the Ethernet network. STP-based technologies use this characteristic to be able to satisfy connection requests, relying on long term monitoring and reconfiguration of the network in order to administrate and coordinate the set up of each tree making use of the different VLANs to route their traffic. An STP-based implementation consists of an Ethernet network plus components to the end host (called network controllers) and/or a centralized or distributed manager to monitor and reconfigure the network. Even thought different implementations have been proposed, in all

of them, the routing problem involves determining for each tree both the nodes and links that it uses as well as the traffic routed using its VLANID.

Related work in the study of STP based technologies can be classified into two categories: a) work proposing a specific STP-based implementation and b) work studying the routing problem in STPbased technologies.

3.2.1 STP-based implementations

Several implementations of STP based technologies have been proposed. One of these implementations is the viking architecture proposed in [SGNC04]. The viking architecture allows to route traffic between source and destination nodes using several spanning trees, additionally, it also allows to support protection by routing backup traffic through assigned backup trees. The VLAN tag selection of the packets is performed by end-hosts instead of the switches, meaning that Viking extends the VIDs until the end-hosts. To address the scalability problems of limited VLANs, Viking relies on an algorithm that minimizes the overall number of required spanning-tree instances while maximizing the number of active links. The implementation does not run directly on the switches, instead it consists of two different components: a client, the Viking Network Controller (VNC), which resides on end-hosts, and a centralized manager or Viking Manager (VM), which is located somewhere on the network, e.g., a centralized server. The VNC performs several tasks such as load measurement, VLAN selection and respective VID tagging. The VM is responsible for traffic engineering and for fault tolerance. Additionally, the VM holds a global view of network resources (based upon information fed by the several active VNCs).

Another implementation is proposed by Farkas et al. in [FATW05], unlike viking, Farkas implementation is distributed. In the same manner as viking, Farkas implementation allows to route traffic between source and destination nodes using several spanning trees, as well as supporting protection by assigning backup trees. The architecture implementation runs exclusively on edge nodes of the Ethernet network, which are typically IP routers. It relies on a distributed method for detection of faults of spanning trees proposed in [FAW$^+$06]. Instead of using a centralized manager like in viking, the method utilizes broadcast messages to check whether a spanning tree is alive or not, to decrease traffic and processing load as much as possible. Other architectures and methods with similar characteristics are proposed in [AA05] and [NNM$^+$06].

[DS06] proposes an algorithm that given a set of specific trees, calculates the MSTP parameters required to be defined on the network so that the MSTP protocol builds the trees. Even though this is not a complete implementation, the algorithm allows to generalize the routing problem of STP based technologies regardless of the implementation.

3.2.2 STP-based routing problem

The routing problem of STP-Based technologies can be generalized and stated as follows: given an offline or online routing problem and a maximum number of spanning trees, the problem is to find a set of paths P solving the routing problem in addition to a set of bidirectional trees T, such that $|T| <= maxt$ and every path $p \in P$ belongs to at least one tree in T ($\forall p \in P, \exists t | p \in paths(t)$).

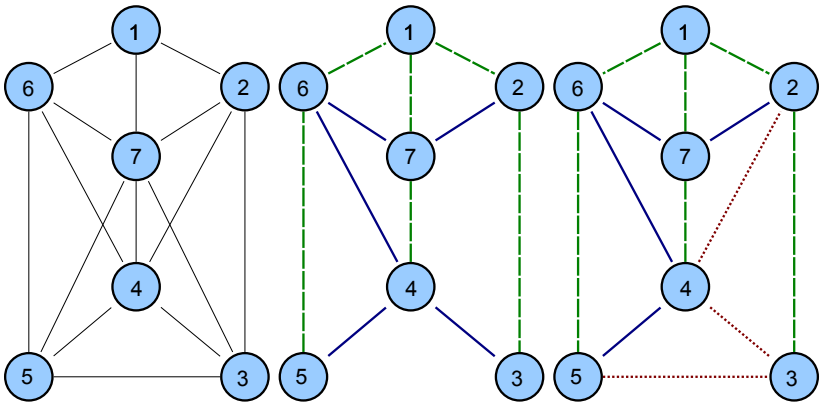
In the online scenario, given the topology of the network, an already accommodated traffic and a new incoming bandwidth request between two nodes, the problem is to find a path across the network that satisfies the bandwidth request. The objective is to decrease the probability of future bandwidth requests being blocked. The complete traffic matrix is unknown and bandwidth requests arrive sequentially.

Several algorithms have been proposed to solve the online routing scenario of STP-based technologies. M. Ali et al. [AA05] propose a heuristic aiming to minimize bandwidth reservation on links in the network. In [SGNC04] an algorithm that minimizes the overall number of required spanning-tree instances while maximizing the number of active links is proposed. The algorithm supports protection by designating working and backup trees for each connection request. In [FATW05] the problem is divided and solved in two parts. They proposed a heuristic to calculate a set of spanning trees for protecting a given topology, for then assigning the traffic to the pre-calculated spanning trees.

The offline routing problem of STP-Based technologies can be stated as follows: Given a maximum number of trees $maxt$, a network graph $G = (N, E)$ and a traffic matrix $TM = N \times N$, where N is the set of nodes, and E the set of links. The problem is to find a set of undirected trees T ($|T| <= maxt$) and accommodate the traffic described by TM, such that the traffic is routed through the paths given by T. The main objective is to maximize the accommodated traffic. If protection is supported then the traffic matrix specifies working and backup traffic that have to be accommodated.

The routing problem can be illustrated using the following example. Given the topology described on Figure 3.8a, with links with a capacity of 10 units of traffic. The traffic matrix TM describes 10 units of traffic to be routed between the pairs (1,2), (1,7), (2,4), (2,7), (3,2), (3,7), (4,3), (4,5), (5,3), (5,6), (5,7), (6,1), (6,4), (7,4), (7,6). If $maxt = 2$, then the optimal set of trees $|T|$ that can be used to accommodate the maximum amount of traffic is as described in Figure 3.8b. In the figure each line color and style represents a tree, note that in the case of $maxt = 2$ the maximum accommodated amount of traffic is 140. In the case of $maxt = 3$ (Figure 3.8c) then all the 150 units of traffic can be accommodated as more links can be used. This example illustrates how the maximum number of trees ($maxt$) limits the performance of STP based implementations. This limitation is one of the main drawbacks of STP based technologies when compared with label-based forwarding technologies. It is important to note that if $maxt = \inf$, then the optimal solution of the routing problem would be equivalent to using label-based forwarding technologies. Given an offline routing problem, the minimum number of trees required to optimally route all the maximum amount of traffic described by TM is defined as $mint$. In the previous example, $mint = 3$, this is because 3 is the minimum number of needed trees required to route the maximum amount of traffic (150 units). Additionally, if $opt(TM, G, maxt)$ is the optimal solution of the offline routing problem, then $opt(TM, G, mint + 1) = opt(TM, G, mint) > opt(TM, G, mint - 1)$.

Related work in the offline routing scenario of STP based technologies includes the study done by A. F. De Sousa [DS06], where an algorithm applicable to the offline scenario is proposed for MSTP that supports load balancing with protection and recovery. An ILP that solves the offline routing scenario, given a set of spanning trees, is proposed by J. Qiu [QMCL08]. The ILP returns for each element of the traffic matrix, a single working and backup spanning tree



(a) 7 node Topology (b) Trees with $maxt = 2$ (c) Trees with $maxt = 3$

Figure 3.8: STP-based routing problem example

to route the demanded amount of traffic. For calculating the set of spanning trees the authors use a heuristic.

For each pair of nodes (s, d), where $TM(s, d) > 0$, we refer to a commodity $c \in C$ such that the requested bandwidth of the commodity $BW(c) = TM(s, d)$ and the destination and source of c are s, d, respectively.

The proposed model by J. Qiu consists of the following indices:

- i,j,n,m,u,v For representing nodes in the network.
- c a commodity given by TM.

And the following parameters:

- (N, E) A network graph with node set N and edge set E.
- K number of established spanning trees.
- TM traffic matrix.
- $C_{(i,j)}$ Capacity of a link.
- BW_c set of requested bandwidths given by TM.
- $S_{(c,i)}$ is set to 1 if node i is the source of commodity c, -1 if it is the destination and 0 otherwise.
- $P^k_{i,j}$ Path from node i to node j on spanning tree k.
- $P^k_{i,j}(n)$ nth hop of path from node i to node j on spanning tree k.

The variables used in the model are the following:

- $r^{(n,m)}_{(i,j)}$ reserved spare capacity on link (i, j) for failure of link (n, m).
- $w_{(i,j)}$ working traffic on link (i, j).

- b_c^k binary variable, 1 if commodity c uses spanning tree k.
- $a_{c,(i,j)}^k$ binary variable, 1 if commodity c uses spanning tree k as backup upon failure of link (i,j).

The objective function is to accommodate as much bandwidth as possible through the entire network.
MAXIMIZE:

$$\sum_{c,k} BW_c \cdot b_c^k \tag{3.1}$$

SUBJECT TO:

$$\sum_{c,k} b_c^k = \{0,1\} \qquad \forall c \tag{3.2}$$

$$\sum_{\substack{k,(i,j)\in P_{i,m}^k \\ |S_{(c,m)}=-1}} a_{c,(i,j)}^k = 0 \qquad \forall c,(i,j) \in E \tag{3.3}$$

$$\sum_{\substack{k,(i,j)\in P_{i,m}^k \\ |S_{(c,m)}=-1}} a_{c,(i,j)}^k = \sum_{\substack{k,(i,j)\in P_{n,m}^k \\ |S_{(c,m)}=-1, S_{(c,n)}=1}} b_c^k \qquad \forall c,(i,j) \in E \tag{3.4}$$

$$w_{(i,j)} = \sum_{\substack{c,k,(i,j)\in P_{n,m}^k \\ |S_{(c,m)}=-1, S_{(c,n)}=1}} BW_c \cdot b_c^k \qquad \forall (i,j) \in E \tag{3.5}$$

$$r_{(i,j)}^{(n,m)} = \sum_{\substack{c,k,(n,m)\in P_{u,v}^k, k', \\ (i,j)\in P_{n,v}^{k'} \setminus P_{u,v}^k \\ |S_{(c,v)}=-1, S_{(c,u)}=1}} BW_c \cdot a_{c,(n,m)}^{k'} \cdot b_c^k \qquad \forall (i,j),(n,m) \in E \tag{3.6}$$

$$w_{(i,j)} + r_{(i,j)}^{(n,m)} + r_{(i,j)}^{(m,n)} \leq C_{(i,j)} \qquad \forall (i,j),(n,m) \in E \tag{3.7}$$

Constraint 3.2 ensures that a connection is either rejected or assigned a VLAN ID, and traffic splitting to multiple spanning trees is not allowed. Constraint 3.3 and 3.4 ensure that only one backup spanning tree is selected for each link along the primary path of the connection. Constraint 3.5, 3.6 and 3.7 ensure that the spare capacity reserved for restoration on each link plus the working traffic do not exceed the link capacity. In addition to the ILP in [QMCL08] a heuristic is also proposed. The performance of both the ILP and the heuristic are evaluated and compared.

In [SMY00] a model to create Multipoint-to-Point LSPs is proposed. Even though a Multipoint-to-Point LSP is a tree shaped connection, the model cannot be used for this problem due to the fact that multipoint-to-point trees are restricted to having the source of the connection as the root. Additionally, the model does not support protection. The same applies for the heuristics proposed in [BGN03].

3.3 Chapter remarks

In this chapter the related work to the contributions of this thesis is summarized. As stated in the previous chapter, one of the main contributions of this thesis

is to study and compare label space usage for ELS and PBB-TE architectures. In this chapter label space usage concepts as well as related work in the use of techniques to improve label space usage in several label based forwarding technologies have been introduced. The considered techniques are label merging, asymmetric tunnels and asymmetric merged tunneling. Both ELS and PBB-TE use a different label size and scope than the technologies considered in the related work. In the case of ELS, even though it uses the same scope as MPLS, only label merging is supported as the technology does not allow to stack labels (which is required for both asymmetric tunnels and asymmetric merged tunneling). In the case of PBB-TE, none of the techniques are supported as it uses a different label scope. Additionally, all the previous studies aim at reducing or minimizing the total number of forwarding states, however the new label size and scope of PBB-TE and ELS limits the maximum number of used labels. Therefore one of the main objectives of this thesis is to study if PBB-TE and ELS label space is scalable as well as how can the existing techniques be applied and adapted to improve each technology label space usage. Another important remark is that none of the previous studies in any of the existing label based forwarding architectures analyzes the impact of the topology characteristics on label space usage. Consequently, another of the main objectives of this thesis is to analyze how topology characteristics affect the number of states and the number of labels needed (relevant for label exhaustion), considering the techniques to improve label space usage available for carrier Ethernet technologies.

This chapter also introduced related work proposing and studying the Carrier Ethernet STPbased technologies. Three implementations have been introduced, and the routing problem of STPbased technologies generalized and formalized. An overview of the related work studying issues of the different implementations as well as the routing problem is presented. Despite all the studies that have been performed for the STPbased technologies, their performance is always compared either among themselves or against the use of basic native Ethernet protocols. To the best of our knowledge, there are no studies comparing label based technologies with STP based. S. Ilyas et al. in [INB+07] present a simulation study of label-based approaches that compares the performance of label-based forwarding approaches against the use of native Ethernet protocols. However, it does not consider the use of multiple spanning trees or any of the approaches referenced in this section for single spanning tree. Additionally there are not any studies that can determine when label based forwarding technologies have to be used instead of STP based. Therefore, one of the main objectives of this thesis is to calculate optimal performance of STP based technologies and compare them with label based forwarding technologies to be able to determine, given a specific scenario, which approach to use.

Chapter 4

Label Space Usage in Carrier Ethernet*

In Chapter 2, Ethernet VLAN-Label Switching (ELS) and Provider Backbone Bridges - Traffic Engineering (PBB-TE) have been introduced. Both technologies improve control and forwarding components by implementing label-based forwarding. The label encoding of each technology limits the number of LSP that the technology supports. Additionally, a forwarding technology is said to experience label scalability issues when, in order to satisfy a capacity request, there is enough capacity to create a new data path but there are not enough free labels on at least one link traversed by that data path.

Given that ELS and PBB-TE use a different label size and scope than previous label based forwarding architectures (such as MPLS which uses 20 bits and per link scope label), in addition to not allowing label stacking, both architectures may be subject to label scalability issues. Given that in carrier networks scalability is a main requirement, this chapter focuses on the study and improvement of label scalability for both architectures. For this purpose, the applicability of existing techniques and studies (explained in Chapter 3) that can be used to overcome or reduce label scalability issues is evaluated for both architectures. After this, a new routing algorithm that improves label space usage for ELS is proposed. For PBB-TE, the label reutilization technique is formalized and the complexity of its optimal use analyzed. Additionally, the influence of the topology characteristics on label space usage is analyzed and used to compare the performance of the two technologies. Finally, chapter conclusions are given.

4.1 ELS label scalability

In a scenario in which labels have an interface scope, the size of the label space limits the number of LSPs that can be forwarded in each link. In ELS, in each link a maximum of 4,096 (2^{12}) LSPs, can be forwarded. In MPLS the maximum is 1,048,576 (2^{20}) without considering stacking.

In MPLS, label size is not considered as a routing limitation like link capacity. However, in ELS, this aspect must be taken into account. Given that Ethernet VLAN-labels have a significantly smaller size and intermediate nodes, i.e. E-

LSRs are not capable of label stacking, it is possible that label space on certain links may have been exhausted before the full capacity of that link has been provisioned. In other words, the label size limitation could represent a new routing constraint, in addition to link capacity. To illustrate this constraint, let us consider the following example. In a carrier network, with an average link capacity of 10Gb/s, it could be said that the acceptable minimum bandwidth for each bandwidth request is equal to or higher than 1Mb/s given that traffic is being aggregated. In this network, given that the minimum bandwidth is 1Mb/s, the maximum number of LSPs that could traverse a link is 10,240. This example illustrates how the ELS label size could become a routing limitation (as $10,240 > 4,096$), while for MPLS it is not (as $10,240 < 1,048,576$).

In Chapter 3 several techniques that can be used on label switching architectures to allow several logical connections to share the same label, thus improving label scalability are explained. Among these techniques only label merging is supported by ELS. In addition to label merging, Chapter 3 also describes previous studies in the optimal use of these techniques. The main goal of these studies is to reduce or minimize the total number of forwarding states, in this section the goal is to determine and overcome the limitations of the 12 bit ELS label. In other words, in previous studies label merging is used to reduce the total number of forwarding states, however in ELS, label merging is used to prevent that certain links experience label exhaustion before capacity exhaustion, i.e., label merging is an integral part of the traffic engineering strategy. In this section the performance of label merging applied to ELS is evaluated. However, performance is evaluated by measuring label exhaustion instead of the total number of forwarding states.

4.1.1 ELS performance evaluation

In this section the performance of ELS in the offline and online routing scenarios is evaluated. In the online scenario, given the topology of the network, a set of already established Ethernet LSP and a new incoming bandwidth request between two nodes, the problem is to find a path across the network that satisfies the bandwidth request. A common objective is to decrease the probability of future bandwidth requests being blocked. In the offline scenario, given the same topology of the network and a [source-destination] matrix describing the entire network traffic, the problem is to find a set of paths capable of routing all (or part) of the traffic as described by that matrix. A common objective is to increase the network's overall throughput. In this scenario, the traffic from a given source-destination can be routed by any number of LSP.

In order to effectively obtain meaningful results three topologies of different sizes are considered: Cost266, Germany50, and Exodus. These topologies are described in terms of number of nodes and number of links in Table 4.1. The table also shows the number of nodes chosen as ingress-egress for the connection requests.

Even though ELS labels have link scope, as some switches do not yet support multiple bridging components, simulations using labels with node scope are also considered.

Name	# nodes	# links	# ing-egr nodes	Source
Cost266 (LT)	37	57	14	[IKM03]
germany50	50	88	20	[ea07b]
Exodus(US)	79	147	31	[SMW02]

Table 4.1: Topology descriptions

Online scenario

The existing routing algorithms considered in this set of simulations are the Shortest Path First (SPF), the Constraint Shortest Path First (CSPF), and the Minimum Interference Routing Algorithm (MIRA).

The SPF selects the path with the minimum cost metric. If several paths with minimum cost metric are found, the one with the minimum number of hops is selected. The implemented CSPF selects the path with the minimum TE-Metric (such as delay). If several paths with the minimum TE-Metric are found, the one with the maximum residual capacity is selected. If there are several with the maximum residual capacity, then the one with the minimum number of hops is selected. The Minimum Interference Routing Algorithm (MIRA) looks for the path with the minimum interference with other source destination pairs. For further information the reader is referred to [KL00].

For all the topologies the link capacity is set to 10Gb/s and for each topology two sets of bandwidth requests serve as input.

- Homogeneous set: each bandwidth request is of 1Mb/s. For this set, the source-destination pairs (ingress-egress node pair) are selected randomly using a uniform distribution.The objective of using this set is to evaluate a scenario where label exhaustion is always reached when not applying any technique. In this case given that all bandwidth requests are of 1Mb/s when not applying any technique, the number of LSPs that will be able to traverse a link will be 4096 (without aggregation or merging each LSP uses one new label), instead of 10240 as it should be given the link capacity.

- Heterogeneous set: the bandwidth of each request is selected among 1Mb/s, 2Mb/s, 10Mb/s, and 20Mb/s. The source-destination pairs and bandwidth of each request are selected randomly using a uniform distribution.

Routes are computed sequentially according to each set of bandwidth requests. The order of the request cannot be altered and accommodated requests are not terminated. For both sets, the algorithms are evaluated with and without the 12 bit label limit. All bandwidth request sets were generated with a number of requests higher than the amount that any of the algorithms can accommodate (this is the reason why none of the algorithms reaches 100% throughput). Each result given below is the average of 10 simulation runs. Confidence intervals of 95 percent were calculated. The algorithms performance is evaluated in terms of the sum of the accommodated bandwidth of all the established LSPs in the network (throughput), and the number of used labels of the link with the highest number out of all the links in the network (maximum number of used labels). The confidence intervals are less than 1 percent for throughput and less than 178 labels for the maximum number of labels. The results of the algorithms without the 12 bit limit are presented in Table 4.2.

Table 4.2: Results without 12 bit label limit

		HmRS		HeRS	
Topology	**Algorithm**	**TH(%)**	**ML**	**TH(%)**	**ML**
Cost266	SPF	56	10240	90	4736
	CSPF	53		90	4992
	MIRA	60		98	3968
Germany50	SPF	60		69	3456
	CSPF	66		70	4480
	MIRA	60		68	3968
Exodus(US)	SPF	76		78	6144
	CSPF	80		81	5632
	MIRA	77		78	5376

HmRS=homogeneous request set, HeRS=heterogeneous request set, TH(%)=throughput and ML=maximum number of used labels.

Results when the 12 bit label limit is applied are presented in Table 4.3. The decrease in throughput (DTH columns) is defined as the difference between the throughput of the algorithm with no label limit (Table 4.2) and the algorithm with the specified limits and techniques.

When considering homogeneous bandwidth requests of 1Mb/s, results show that with the label size restricted to 12 bits per link (labels per link limit, no technique column in Table 4.3), all evaluated algorithms present a decrease in throughput that ranges from 32% (Cost266 with CSPF) to 50% (Exodus with CSPF). With a restricted label size but aggregation and label merging enabled, the decrease in throughput (labels per link limit, agg + merg column in Table 4.3) is almost 0 (except for a 1% decrease for MIRA in Exodus). When the label size is restricted to 12 bits per node (labels per node limit column in Table 4.3), the decrease in throughput ranges from 46% (Cost266 with CSPF) to 69% (Exodus with CSPF) and with aggregation and label merging enabled from 12% (Germany50 with SPF and MIRA) to 22% (Exodus with MIRA). The latter observation applies for all the algorithms evaluated. Results show that aggregation together with label merging overcomes the label size limitation for a link scope when the bandwidth requests are as low as 2.5% of the link capacity (in this case). This is not the case when the labels have a node scope where there are still limitations even with merging.

When considering heterogeneous bandwidth requests, with the label size restricted to 12 bits per link, none of the evaluated algorithms show a decrease in throughput higher than 2%. This is due to the fact that with higher bandwidth demands, full capacity is reached before reaching sparsity of labels. In addition to these results, when aggregation and label merging are applied, the maximum number of used labels decreases considerably, ranging from 48% (from 3456 to 1792 labels, case of Germany50 with SPF) to 57% (from 4480 to 1920 labels, case of Germany50 with CSPF). When the label size is restricted to 12 bits per node, the decrease in throughput ranges from 19% (Germany50 with MIRA) to 29% (Exodus with CSPF), and with aggregation and label merging enabled, from 1% (Cost266 with SPF) to 11% (Exodus with SPF and CSPF).

These results show that when nodes have a per node label space and the bandwidth of the LSP is low in comparison to the capacity (1Mb LSP with

Table 4.3: Results with 12 bit label limit

Homogeneous request set									
Topology	**Algorithm**	labels per link limit				labels per node limit			
		no technique		agg + merg		no technique		agg + merg	
		DTH	**ML**	**DTH**	**ML**	**DTH**	**ML**	**DTH**	**ML**
Cost266	SPF	33	4096	0	3456	48	4096	18	4096
	CSPF	32		0	3584	46		16	
	MIRA	37		0	4096	52		16	
Germany50	SPF	35		0		52		12	
	CSPF	42		0		59		16	
	MIRA	36		0		51		12	
Exodus(US)	SPF	41		0		66		15	
	CSPF	50		0		69		15	
	MIRA	41		1		66		22	
Heterogeneous request set									
Cost266	SPF	1	4096	0	2048	20	4096	1	4096
	CSPF	1	4096	0	2176	21		3	
	MIRA	0	3968	0	1792	22		6	
Germany50	SPF	0	3456	0	1792	23		2	
	CSPF	0	4096	0	1920	26		4	
	MIRA	0	3968	0	2048	19		7	
Exodus(US)	SPF	2	4096	0	2944	27		11	
	CSPF	0	4096	0	2560	29		11	
	MIRA	0	4096	0	2688	26		7	

DTH(%)=decrease in throughput and ML=maximum number of used labels.

10Gb of capacity), a 4096 label value space can be a limitation.

Offline scenario

For the offline routing scenario, as in [AT03] an integer linear program (ILP) modeling the multi-commodity flow problem was evaluated. The model has as an objective function to maximize the total accommodated bandwidth expressed by:

$$\sum_{i,j,c} f(i,j,c) \qquad \forall i,j \in N, c \in C | i = S_c \tag{4.1}$$

Where N is the set of nodes, C is the set of commodities for which S_c is the source of the commodity c, and $f(i,j,c)$ is the flow of commodity c through the edge (i,j). For each solution of the model, the maximum number of used labels per link and maximum number of used labels per node were calculated when using and not using label merging. The same topologies with the same capacities are considered. The traffic matrix was generated following a uniform distribution with the requirement that it describes more traffic than what can be accommodated in each network.

Topology	no merging		merging	
	MLL	**MLN**	**MLL**	**MLN**
Cost266	141	134	47	47
Germany50	120	114	30	28
Exodus(US)	135	92	24	22

MLL=maximum number of used labels per link and MLN=maximum number of used labels per node.

Table 4.4: Offline Results

Results obtained when using the off-line scenario with the topologies described in Table 4.1 are presented in Table 4.4. The highest maximum number of utilized labels was 141 which is very low compared to the 3955 unused labels (around 3.5%). This result shows that even without merging, for the offline scenario a 4096 label value space is not a limitation. When comparing offline and online results, in the offline scenario given that the full traffic matrix is known and splittable, LSP tend to be of higher bandwidth than in the online scenario. In this case, in the offline scenario the lowest bandwidth LSP given by the ILP was higher than 50Mb, given that links are of 10Gbs capacity this explains why there were not limitations. On the contrary, in the online scenario it is possible to have LSP as low as 1Mb, creating the possibility that a 4096 label is a limitation.

4.1.2 A novel online routing algorithm based on CSPF

Due to the decreases in throughput presented in Section 4.1.1 when nodes have a per node label space and to the fact that the capacity of an aggregation network can be higher than 10Gb/s (e.g. 100Gb/s), an online routing algorithm designed to improve label space usage is proposed in this section.

The Constraint Shortest Path First (CSPF) algorithm considers a number of n given constraints $\{C_1, C_2, ..., C_n\}$. The algorithm at first looks for a path based on the first constraint C_1. If more than one path is found, the following constraint is used, in this case C_2. In a general case, if several paths based on C_i are found, C_{i+1} is applied. The process continues until only one path is found or until all of the constraints ($i = n$) have been applied. The most usual implementation of the CSPF (implemented in Section 4.1.1) uses three constraints {TE-METRIC, max residual capacity, hop count}.

In order to define a new routing scheme that uses label size and network resources efficiently, new routing metrics designed for optimizing label space usage are introduced.

Given a link l and a node d, we denominate the $MergD(l, d)$ function "Merging Degree", which we define as the maximum number of LSPs that are forwarded through l, end at node d and have the same label assigned on link l. The minimum value of the merging degree is 0, i.e. there are no LSPs routed through l with d as their destination. An illustrative example is presented in Figure 4.1. The figure shows several LSPs that are routed through a link (l), each line represents a different LSP. LSPs with the same line style have the same label assigned. The first four LSPs end at node d, three having label A and one having label B assigned; therefore, $MergD(l, d) = 3$ in this case.

Figure 4.1: $MergD$ example

The merging degree ($MergD(l, d)$) function represents a routing metric just as TE-METRIC does. Additionally the number of unused labels on the link represents a routing metric in a similar way as the residual capacity does. The proposed online routing algorithm considers the metrics shown in Table 4.5.

Table 4.5: Metrics comparison

Previous Constraints	**New constraints**
TE-METRIC (C_{a1})	MergD (C_{b1})
Maximum residual capacity (C_{a2})	number of unused labels (C_{b2})
number of hops (C_3)	number of hops (C_3)

Two strategies are defined for the algorithm. The first strategy (hCSPF) intends to maintain a homogeneous distribution of the merging degree, minimizing its variation between the links of the network. This is done with the objective of distributing the load between links in order to prevent that certain links of the network suffer from label exhaustion. The second strategy (mnCSPF) intends to perform routing so as to improve the re-use of common segments (and thus take benefit of label re-utilisation) from intermediate nodes to the common destination shared by a set of LSPs. Both strategies are specifically designed to

make a better use of the label merging technique.

For both strategies the algorithm input is the network topology, a set of established LSPs, and a connection request from node s to node d of D units of bandwidth. The algorithm also has as input two weights w_a and w_b, where $0 \leq w_a \leq 1$, $0 \leq w_b \leq 1$ and $w_a = 1 - w_b$. Each strategy calculates a set of paths S between s and d. When S has been calculated, a path is selected based on the maximum residual capacity and maximum number of unused labels, considering the given weights w_a and w_b. More details about this procedure are explained at the end of this section.

Both strategies calculate a path by running a CSPF using the constraints illustrated in the "previous constraints" column of Table 4.5. The obtained path is inserted into the set S.

The hCSPF strategy calculates a second path using a constraint relaxation method that consists of the following four steps:

1. Like the regular CSPF, it prunes all the links with a residual capacity smaller than D.

2. It analyzes all the $MergD$ of all the links of the network and obtains the maximum (x) and minimum merging (y) degree values, then the objective merging degree $obj = \frac{x+y}{2}$ is calculated.

3. It prunes all the links l of the network with a $MergD(l, d) > obj$.

4. Finally the path with the lowest $MergD$ variation is selected. If it exists, the path is inserted into S, otherwise (s and d are not connected) the path with the lowest $MergD$ variation, taking into account the topology with the links pruned on the previous step, is selected. If a new path is found then it is inserted into S.

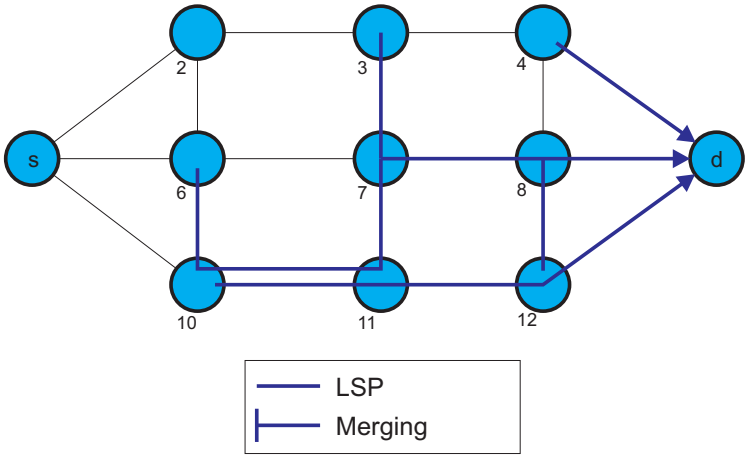
The mnCSPF strategy uses the concept of a merging node. A merging node is any node, other than d, through which one, or several, LSPs ending at node d are established. The mnCSPF follows the following steps:

1. Like the regular CSPF, all the links with a residual capacity smaller than D are pruned. The merging nodes of the network are identified based on all the established LSPs ending at d. A merging graph consisting of the merging nodes, the destination node and the links used by the established LSP is built.

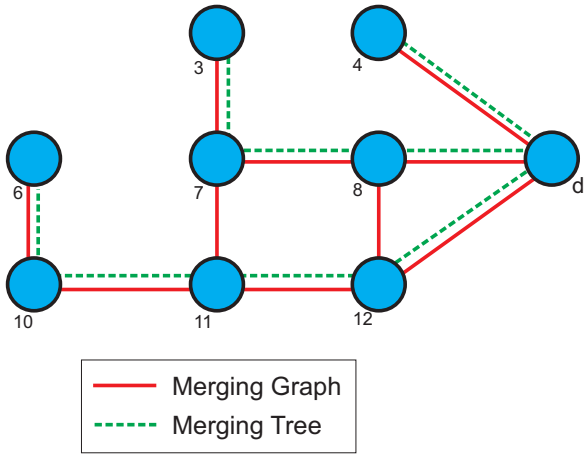
2. A shortest path tree based on the number of hops, going from the merging points to the destination node, is calculated on the merging graph using Dijkstra's algorithm. This tree is called a merging tree.

   An illustrative example of the merging graph and merging tree is presented in Figure 4.2. Given the network topology described in Figure 4.2a, where five LSPs ending at d are established, the five LSPs follow the paths (6,10,7,8,d), (10,11,12,d), (3,7,8,d), (12,8,d) and (4,d), respectively. Merging is performed at nodes 7 and 8. The merging nodes with the merging graph and the merging tree calculated are illustrated in Figure 4.2b. The merging tree is described using dashed lines and the merging graph using continuous ones.

3. Finally, two paths are calculated and inserted into S. The first path is the one with the maximum residual capacity from s to any of the merging nodes joined with the path connecting the merging point with d in the merging tree. The second path is the one with the maximum number of unused labels from s to any of the merging nodes joined with the path connecting the merging point with d in the merging tree.



(a) network topology



(b) Merging Graph with Merging Tree

Figure 4.2: Example of structures used on mnCSPF

Once the set S has been calculated, two paths p_{ul} and p_{rc} are taken from the set, p_{ul} being the path with the highest number of unused labels of the set S and p_{rc} being the path with the highest residual capacity of the set S. One of the two paths is chosen based on the following function:

$$CS(p_{ul}, p_{rc}) = \begin{cases} p_{rc} & \text{if } W_a \cdot \frac{RC(p_{rc})+1}{RC(p_{ul})+1} > W_b \cdot \frac{UL(p_{ul})+1}{UL(p_{rc})+1} \\ p_{ul} & \text{otherwise} \end{cases} \tag{4.2}$$

where $RC(p)$ returns the total residual capacity of path p and $UL(p)$ returns the total number of unused labels of path p.

As the studied problem has two constraints (label usage and bandwidth capacity) for link exhaustion, the weights represent the importance of improving one constraint over the other. The weight should be assigned by the network administrator based on the LSPs expected size in comparison with the capacity of the network links. Equation 4.2 normalizes the impact on the used labels and residual capacity each of the candidate paths has on the network, and by taking into account the assigned weights, it chooses a path. The algorithm and the two strategies are formally described in Algorithm 1.

Algorithm 1: Proposed algorithm

Input: A graph G, source node s, destination d, D units of Bandwidth and weights w_a,w_b
Result: A path pa

$p \leftarrow$CSPF(G, s, d, D)
$S \leftarrow \{p\} \bigcup S$
$S \leftarrow$ mnCSPF(G, s, d, D, S) or $S \leftarrow$ hCSPF(G, s, d, D, S)
$p_{ul} \leftarrow$ ExtracHighNumLabels(S)
$p_{rc} \leftarrow$ ExtracHighResCapacity(S)
$pa = CS(p_{ul}, p_{rc})$

Function `hCSPF`(G, s, d, D, S)

$G' \leftarrow$ Prune(G, D)
calculate maximum (x) and minimum merging (y) degree values
$objM \leftarrow \frac{x+y}{2}$
$G'' \leftarrow$ Prune$(G', objM)$
$p \leftarrow$lowMergDVar(G'')
if $p = null$ **then**
  $p \leftarrow$lowMergDVar(G')
end
$S \leftarrow S \bigcup \{p\}$
return S

Function `mnCSPF`(G, s, d, D, S)

$G' \leftarrow$ Prune(G, D)
$MG \leftarrow$ CalcMergGraph(G')
$MT \leftarrow$ Dijkstra(MG, d)
$p_1 \leftarrow$ path with maximum residual capacity from s to any of the merging nodes
$p_1 \leftarrow p_1 \bigcup$ getPath$(MT, lastNode(p_1), d)$
$p_2 \leftarrow$ path with maximum number of unused labels from s to any of the merging nodes
$p_2 \leftarrow p_2 \bigcup$ getPath$(MT, lastNode(p_2), d)$
$S \leftarrow S \bigcup \{p_1\} \bigcup \{p_2\}$
return S

4.1.3 Algorithm performance

The performance of the algorithm is evaluated in terms of its decrease in throughput in comparison with the CSPF without label limits. First the algorithm schemes are implemented in the same scenario of Section 4.1.1. Only the case in which labels have a node scope is evaluated because it had a decrease in throughput even when label merging and aggregation were used. Based on preliminary simulations, the weights of the mnCSPF and hCSPF were set to $(0.5, 0.5)$.

Results are presented in Table 4.6. For the heterogeneous and homogeneous bandwidth requests, the proposed algorithm had either higher or equal performance as the CSPF+ (with labels per node limit and merging and aggregations applied). For the heterogeneous bandwidth requests, the decreases in throughput are overcome by the mnCSPF+, except for the Exodus topology where the scheme presented a decrease of 3%, which is less than half of the decrease of CSPF+. For the homogeneous bandwidth request, both schemes presented decreases in throughput, less than the CSPF+. In the Cost266 and Exodus(US) topologies mnCSPF+ presented the best performance with a decrease of 8% and 9%, respectively, which is half and 2/3 of the decrease of CSPF+. In the Germany50 topology hCPSF+ presented the best performance with a decrease of 8%, which is half of the decrease of CSPF+.

Table 4.6: Algorithm Decreases in throughput(%) with 10Gb/s links and node scope

Heterogeneous request set			
Algorithm	**Cost 266**	**Germany50**	**Exodus(US)**
CSPF+	3	4	11
mnCSPF+	-1	0	3
hCSPF+	2	7	7
Homogeneous request set			
CSPF+	16	16	15
mnCSPF+	8	13	9
hCSPF+	14	8	15

+ the algorithm uses merging.

Additionally, in order to further evaluate the performance of the proposed algorithm, a scenario where the capacity of the links is 100Gb/s and labels have a link scope is considered. The same three topologies are used and for each one a homogeneous set of bandwidth requests of 1Mb/s serves as input. The CSPF, mnCSPF and hCSPF with the label limit and applying label merging and aggregation are evaluated in terms of used labels and decreases in throughput. For all the algorithms the decrease in throughput is calculated based on the throughput of the CSPF without any label limit. For the mnCSPF and hCSPF, based on the previous results and preliminary simulations, their weights were set to $(0.3, 0.7)$.

Table 4.7 presents the calculated decreases in throughput (%) and maximum number of labels. The table shows that the CSPF, when used with merging and aggregation, can have a decrease in throughput of 7%. On the contrary, the hCSPF has a decrease of, at most, 5% and the mnCSPF did not have a decrease

higher than 1%.

Table 4.7: Decreases in throughput (%) and maximum number of labels with 100Gb/s links and link scope

	Cost 266		Germany50		Exodus(US)	
Algorithm	**DTH**	**ML**	**DTH**	**ML**	**DTH**	**ML**
CSPF+	1	4096	7	4096	7	4096
mnCSPF+	1	4096	1	4096	1	4096
hCSPF+	1	4096	5	4096	5	4096

+ the algorithm uses merging.

Figures 4.3,4.4 and 4.5 show the average and maximum number of labels in terms of the offered load of the network. The offered load can be defined as the amount of connection request that have been received (either established or rejected). The figure shows the values until maximum load is reached.

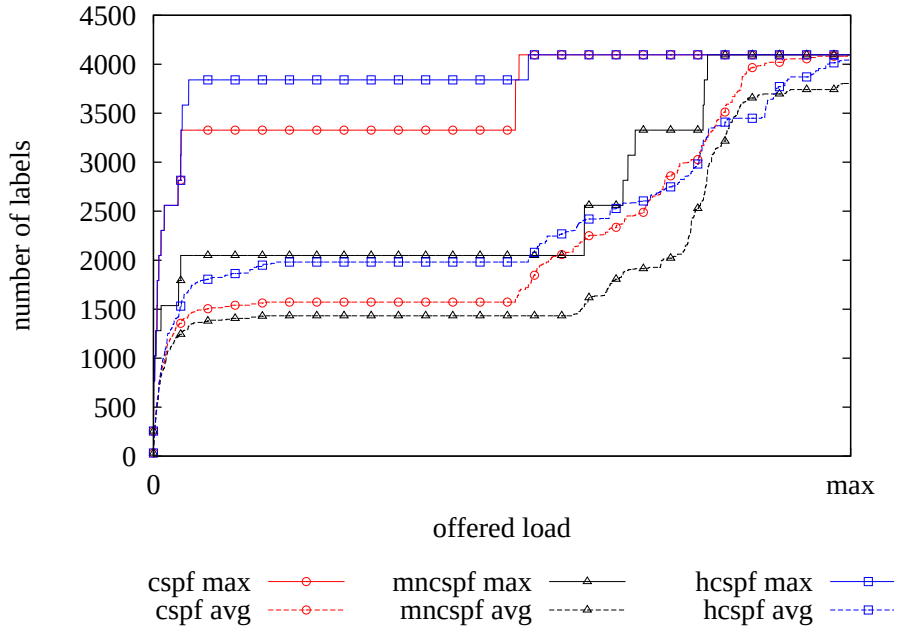


Figure 4.3: Maximum and average number of labels with 100Gb/s links and link scope for Cost266

For the case of the Cost266 topology (Figure 4.3) when the CSPF maximum number of labels reaches 4096 (which is the architecture limit), the mnCSPF and hCSPF have 2048 (50% of 4096) and 3840 (93%) maximum number of labels, respectively. When the CSPF average number of labels reaches 4096 the mnCSPF and hCSPF have 3804 (92%) and 4051 (98%) average number of labels respectively.

For the case of the Germany50 topology (Figure 4.4), when the CSPF maximum number of labels reaches 4096 the mnCSPF and hCSPF have 2560 (62%)

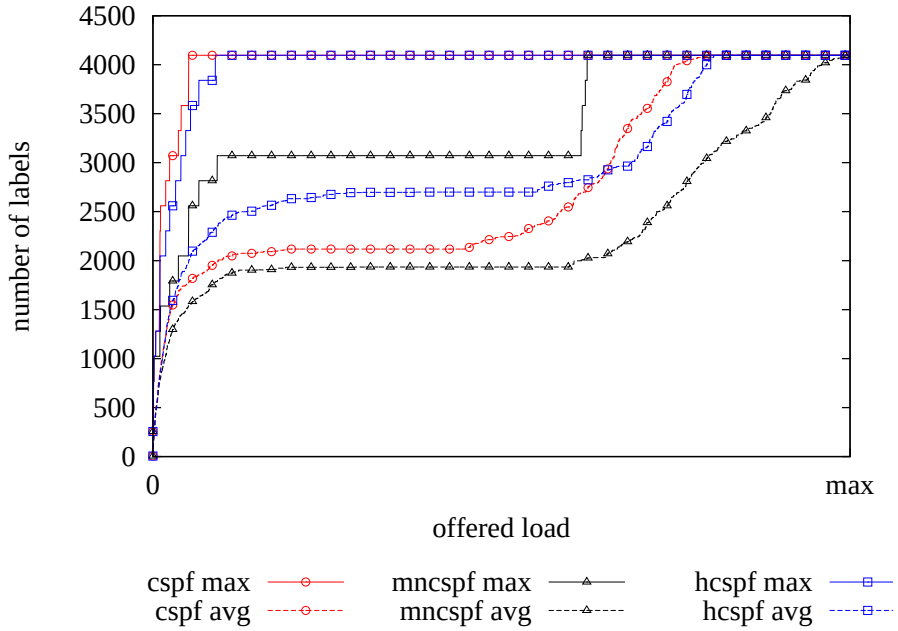


Figure 4.4: Maximum and average number of labels with 100Gb/s links and link scope for Germany50

and 3328 (81%) maximum number of labels, respectively. When the CSPF average number of labels reaches 4096, the mnCSPF and hCSPF have 3074 (75%) and 4037 (98%) average number of labels, respectively.

For the case of the Exodus Topology (Figure 4.5), when the CSPF maximum number of labels reaches 4096 the mnCSPF and hCSPF have 2304 (56%) and 3328 (81%) maximum number of labels, respectively. When the CSPF average number of labels reaches 3895 (which is the maximum reached), the mnCSPF and hCSPF have 3688 (94% of 3895) and 4015 (103%) average number of labels, respectively.

In summary, the algorithm that was able to accommodate more traffic before reaching 4096 (ELS limit) average and maximum number of labels, is mnCSPF. Results also show that the degree of label sparsity depends more on the ratio between the size of the LSPs and the links capacity, than the size of the network.

4.2 PBB-TE label scalability

In the case of PBB-TE, labels are globally unique and encoded on both B-VID and B-DA fields. Therefore on PBB-TE a maximum of 4096 LSPs per destination MAC address can be created. Given that the label scope of PBB-TE is different from previous technologies, it is important to evaluate if its label scope together with its size can present label scalability limitations. For this case the number of supported LSP is independent of the number of links of the topology.

In Chapter 3 several techniques that can be used on label switching architec-

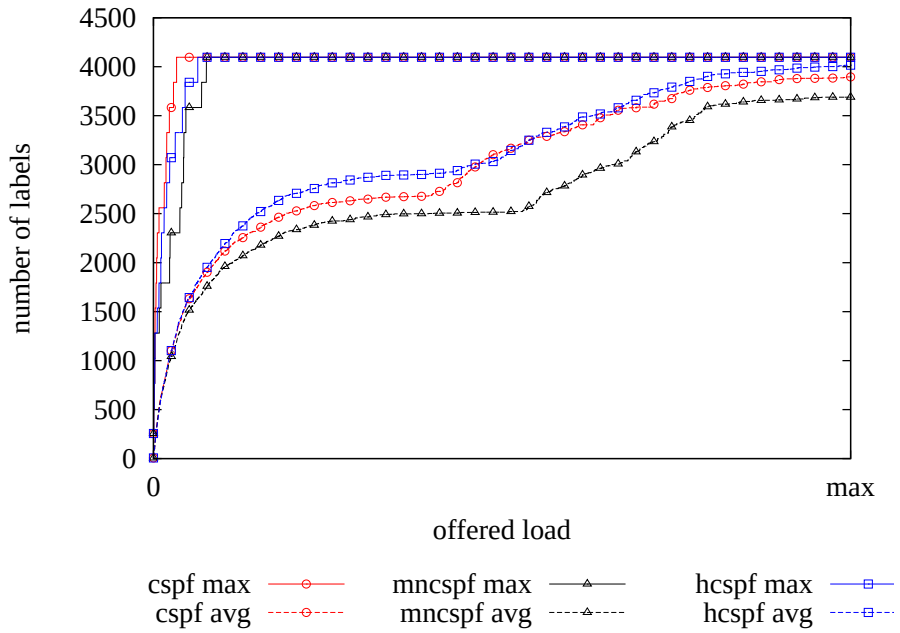


Figure 4.5: Maximum and average number of labels with 100Gb/s links and link scope for Exodus(US)

tures to allow several logical connections to share the same label, thus improving label scalability are explained. Out of these techniques only inverse trees are supported by PBB-TE.

4.2.1 Label reutilization

For PBB-TE, another technique that could be used to improve label space, besides inverse trees, is label reutilization. In this section the technique is formalized and the complexity of optimally applying it shown. The technique consists of assigning the same label to LSPs that are fully link disjoint. In the example of Figure 4.6, where three LSPs are established, in a labels per destination scenario, labels could be assigned as follows: label A to $LSP1$, label B to $LSP2$, and label C to $LSP3$. The number of labels used would be 3 for node 6. However, if label reutilization is used, label A can be assigned to $LSP1$ and $LSP2$, and the number of used labels would therefore be two for node 6. Label reutilization does not reduce the number of forwarding states, it only reduces the number of labels used per destination.

Complexity

When label reutilization is applied with or without aggregation, assigning labels to the LSP is not trivial. The problem of optimally assigning the labels for a set of LSPs routes, considering label reutilization can be formulated as follows; Given a set of Paths P and a set of labels L, the problem is to assign each path a label $label_p = l, \forall p \in P, l \in L$ such that $label_{p1} = label_{p2} \iff$

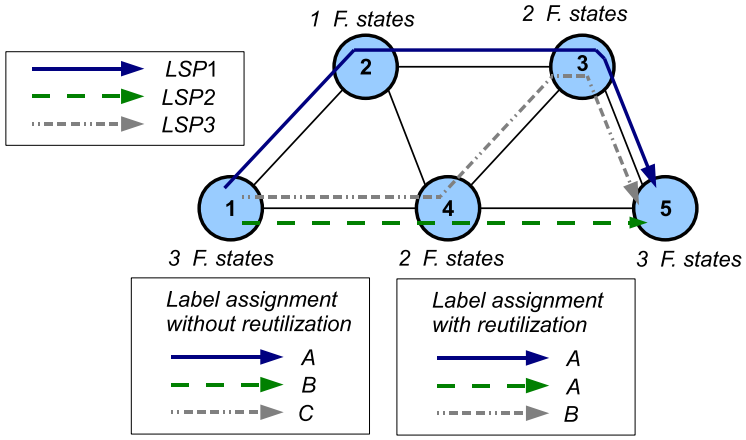


Figure 4.6: Label reutilization example

$links(p1) \cap links(p2) = \emptyset$ (label reutilization), the objective is to minimize the number of used labels which is equal to $|\{l \in L | \exists p \in P, label_p = l\}|$. This problem is NP-complete. For proving its complexity we show that solving the Static Wavelength Assignment problem (which has been proved to be NP-complete in [CGK92]) would also solve the label assignment, and that solving the label assignment would also solve the SLE. Define a graph $G(V, E, W)$ where $|W| = |L|$ and the set of Lightpaths $LI = li | \exists p, p \in P, Links(li) = links(p)$ where $\forall p \in P, \exists li, Links(li) = links(p)$. Then finding a feasible solution of the SLE given G, W, LI will also yield a feasible label assignment solution. In the same manner, define a set of labels L where $|W| = |L|$ and a set of paths $P = p | \exists li, li \in LI, links(p) = Links(li)$ where $\forall li \in LI, \exists p, Links(p) = links(li)$. Then finding a feasible solution to the label assignment problem given P, L also yields a feasible SLE solution.

Given that the problem of label assignment is NP-complete, for evaluating the number of labels when using label reutilization in an online routing scenario, two heuristics called first fit and greedy assignment are implemented.

First fit algorithm

Given the sets of paths, P, and labels, L, the first fit label assignment algorithm takes each path sequentially and assigns the first available label. This is the most basic heuristic for this type of problem and it can be applied in any routing scenario.

Greedy assignment algorithm

Given the sets of paths, P, and labels, L, the greedy algorithm calculates the largest set of Paths that can share a label and assigns them a new label. Then the procedure is repeated with the rest of the unlabeled paths until all paths have a label assigned. This scheme can be applied in online routing scenarios where labels can be reassigned each time a new demand arrives. For the context of this document it is assumed that this is feasible.

4.2.2 PBB-TE performance evaluation

The performance of PBB-TE is only evaluated in the online routing scenario. Offline scenario is not considered, as based on the results of Section 4.1.1, it can be concluded that PBB-TE does not present limitations in this scenario. Online routing is evaluated using the same topologies and routing algorithms of Section 4.1.1. This means that for all the topologies the link capacity is set to 10Gb/s and for each topology the homogeneous and heterogeneous sets of bandwidth requests serve as input.

Five cases are analyzed: 1) ELS without aggregation and merging, 2) ELS with aggregation with merging (referred to as ELS+), 3) PBB-TE without aggregation and VLAN reutilization, 4) PBB-TE with aggregation and VLAN reutilization when applying first fit (FF) algorithm (referred to as PBB-TE+), 5) PBB-TE with aggregation and label reutilization when applying greedy algorithm (GA) (referred to as PBB-TE++) and 6) PBB-TE with aggregation, inverse trees and label reutilization when applying greedy algorithm (GA) (referred to as PBB-TE*).

Routes are generated in the same manner as in Section 4.1.1. The same percentage of confidence intervals and number of simulation runs are considered. The results of the algorithms without any label limit are presented in Table 4.8.

Results of the algorithms with the 12 bit label limit are presented in Table 4.9. The decrease in throughput (DTH columns) is defined as the difference between the throughput of the algorithm with no label limit (Table 4.2) and the algorithm with the specified technology limits and applied techniques.

When considering homogeneous bandwidth requests of 1Mb/s:

- For ELS without label merging and aggregation all the algorithms have a decrease in throughput higher than 50%. When label merging and aggregation are applied, there is no decrease in throughput except for the CSPF in Germany50 and the MIRA in Exodus having only a 1% decrease.
- For PBB-TE without any technique all the algorithms have a throughput of 24%, in other words, throughput decreases by 64% up to 76%. When aggregation and label reutilization with the First Fit (F.F) heuristic is applied, the decrease in throughput is considerably lower varying from 10% up to 33%. When the Greedy Assignment (G.A.) heuristic is applied, the

Table 4.8: Throughput(%) without any label limit

Topology	Algorithm	HmRS	HeRS
Cost296	SPF	92%	90%
	CSPF	88%	90%
	MIRA	100%	99%
Germany50	SPF	91%	69%
	CSPF	100%	70%
	MIRA	91%	68%
Exodus(US)	SPF	95%	78%
	CSPF	100%	81%
	MIRA	96%	78%

HmRS=homogeneous request set, HeRS=heterogeneous request set

Table 4.9: Decrease in throughput(%)

Homogeneous request set							
Topology	**Algorithm**	**ELS**	**ELS+**	**PBB-TE**	**PBB-TE+**	**PBB-TE++**	**PBB-TE***
Cost296	SPF	54%	0.5%	68%	33%	22%	1%
	CSPF	53%	0.5%	64%	32%	20%	1%
	MIRA	61%	0.5%	76%	32%	22%	1%
Germany50	SPF	53%	0.5%	67%	25%	13%	0.5%
	CSPF	63%	1%	76%	27%	22%	0.5%
	MIRA	54%	0.5%	67%	13%	7%	1%
Exodus(US)	SPF	51%	0.5%	71%	15%	3%	1%
	CSPF	63%	0.5%	76%	10%	2%	1%
	MIRA	51%	1%	72%	15%	2%	0.5%
Heterogeneous request set							
Topology	**Algorithm**	**ELS**	**ELS+**	**PBB-TE**	**PBB-TE+**	**PBB-TE++**	**PBB-TE***
Cost296	SPF	1%	0.1%	1%	0.1%	0.1%	0.1%
	CSPF	1%	0.1%	2%	2%	2%	0.1%
	MIRA	1%	1%	6%	5%	5%	1%
Germany50	SPF	0.1%	0.1%	1%	0.1%	0.1%	1%
	CSPF	0.1%	0.1%	5%	3%	3%	1%
	MIRA	1%	0.1%	5%	5%	5%	0.1%
Exodus(US)	SPF	2%	0.1%	4%	0.1%	0.1%	0.1%
	CSPF	0.1%	0.1%	4%	0.1%	0.1%	0.1%
	MIRA	0.1%	0.1%	2%	1%	1%	0.1%

decrease in throughput is even lower varying from 2% up to 22%. It can be appreciated that PBB-TE with aggregation and label reutilization, has a performance proportional to the total number of links of the topology. One possible cause is: the higher the node degree the more disjoint paths that can be found thus VLAN reutilization is more effective. When aggregation, label reutilization and inverse trees are applied, the decrease in throughput is not higher than 1%.

When considering heterogeneous bandwidth requests:

- For ELS, without label merging and aggregation all the algorithms have a decrease in throughput lower than 2%. When label merging and aggregation are applied, there is no decrease in throughput except for the MIRA in Exodus having only a 1% decrease.
- For PBB-TE without aggregation and label reutilization throughput decreases by 1% up to 6%. When aggregation and label reutilization are applied, the decrease in throughput varies from 1% up to 6% regardless of the implemented heuristic, as both heuristics presented the same performance. When aggregation, label reutilization and inverse trees are applied, the decrease in throughput is not higher than 1%.

In summary, for both PBB-TE and ELS, applying the available techniques significantly improves label space usage in the two considered bandwidth request sets. Results also show that the highest decrease in throughput of the two request sets was 70%.

4.3 Chapter remarks

In this Chapter the problem of label scalability in carrier Ethernet has been studied. Two technologies, Provider Backbone Bridges - Traffic Engineering and Ethernet VLAN-Label Switching are considered, where the label scope and value space could result in scalability limitations. The bandwidth granularity associated to labels is analyzed as an indicator of the possibility of sparsity of labels. Several available techniques that can be used to improve label scalability are reviewed and analyzed. Both online and offline routing scenarios are considered.

Two major contributions of this thesis are presented in this chapter:

- For the online routing scenario, three traditional routing algorithms are implemented and tested in order to measure an upper bound on the decrease in performance given by the label space. For the offline routing problem an ILP is used, and the number of labels needed by the optimal solution is analyzed. This contribution has been published in [CPM08c].

- Further on for ELS, a new online routing algorithm designed to take advantage of label merging is proposed and evaluated as well. Two different merging strategies (hCSPF and mnCSPF) considering the improvement of the label space in conjunction with traffic engineering metrics are proposed and tested. For PBB-TE the VLAN-reutilization technique is formalized and the problem of assigning labels to a set of LSPs when using aggregation and VLAN reutilization is shown to be NP-complete. Since the problem is NP-complete, two label assignment heuristics are evaluated. This contribution has been published in [CPM08a, CPM09a].

Results for the offline routing scenario show that even without applying any technique both technologies do not present label limitations. Results for the online routing scenario for both technologies show that for demands of low granularity (1Mb/ which is the acceptable minimum) are considered, performance degradation can be seen when no label reduction techniques are used. However, in networks with a link capacity of 10Gbs, applying the evaluated and proposed techniques allows performance to be maintained in terms of accommodated traffic load. In other terms, the techniques significantly reduces the probability of exhausting the label space before the corresponding unreserved (link) capacity drops to 0.

Chapter 5

Label Space Dependency on Network Topology*

Previous studies on the improvement of label space usage have been performed for different label based forwarding architectures (see Chapter 3). Depending on the architecture itself, these studies have targeted different objectives. Chapter 4 analyzed label exhaustion for ELS and PBB-TE, results show that for the studied topologies, label scalability issues can be overcome for both technologies.

However, none of the previous studies specifically addresses the impact of the topology characteristics on label space usage. In this chapter the influence of the topology characteristics on label space usage is analyzed based on both the number of states and the number of labels needed (relevant for label exhaustion). The objective is to study how the topology characteristics affect the improvement gained by applying available techniques to improve label space usage. Additionally the study compares the performance of the different available label scopes of carrier Ethernet technologies.

5.1 Analytical study

In order to analyze the relationship between the topology type and the label space, an upper bound for the labels used for different types of topologies is determined. The base topologies considered are the following: line, ring, star, general tree, and full mesh. To determine the upper bound, the maximum number of paths between all the nodes allowed by the topology is assumed to be established. The maximum number of paths is calculated based on the number of nodes (n) in order to then calculate the respective maximum number of paths per link and per destination. The techniques for improving label space usage are not considered in this section because some of them (inverse trees and label reutilization) involve an NP-complete decision problem.

Each topology is represented by a directed graph $G = (N, E)$, where N is the set of vertices and E is the set of links, meaning $|N| = n$. G is also symmetric, therefore if a link exists between vertices (i, j), then one also exists between vertices (j, i).

5.1.1 Tree topologies

G is a tree if any pair of vertices (i, j) can only be connected by exactly one path. In the context of this subsection, we refer to $Path_{i,j}$ for the path that connects the vertices (i, j), which must be unique according to the definition of a tree.

Given the properties of a tree, the total number of paths (TNP) for any tree graph is:

$$TNP(G) = n \cdot (n-1) \tag{5.1}$$

Additionally, the maximum number of paths per destination ($MNPD$) is:

$$MNPD(G) = n - 1 \tag{5.2}$$

In order to determine the maximum number of paths per link, for each link (i, j), we define two sets:

- $S_{(i,j)} = \{k \in N | (i, j) \in Path_{k,j}\}$
- $T_{(i,j)} = \{k \in N | (i, j) \notin Path_{k,j}\}$

Both $S_{(i,j)}$,$T_{(i,j)}$ range from 1 to $n - 1$ and:

- $S_{(i,j)} \cup T_{(i,j)} = N$
- $S_{(i,j)} \cap T_{(i,j)} = \{\}$
- $|S_{(i,j)}| + |T_{(i,j)}| = n$

Based on these two sets, the maximum number of paths that can traverse link $MNPL(i, j)$ will be given by the following function:

$$MNPL(i, j) = |S_{(i,j)}| \cdot |T_{(i,j)}| \tag{5.3}$$

Therefore, the maximum number of paths that can traverse a link for the whole graph ($MNPL(G)$) is given by the function:

$$MNPL(G) = Max_{(i,j)}(|S_{(i,j)}| \cdot (n - |S_{(i,j)}|)) \tag{5.4}$$

The value of $MNPL(G)$ depends on the specific topology of the tree. In addition to the tree topology, line and star topologies are also represented by a tree graph. Based on their specific characteristics, we can only determine the value of $MNPL(G)$, in terms of the number of nodes.

Line topology In a line topology represented by G, the graph consists of a sequence of vertices such that from each one there is an edge to the next vertex in the sequence, given that the sequence is ordered in the following way $\{v_1, v_2, ...v_i, .., v_n\}$, where:

- v_1 its only connected to v_2
- v_n its only connected to v_{n-1} and
- v_i its connected to v_{i+1} and to v_{i-1} where $i \neq 1 \neq n$.

This means that for a line topology $|S_{(i,j)}| = i$ where $1 \leq i \leq n-1$, then MNPL(G) can be expressed as:

$$MNPL(G) = Max_{(i)}(i \cdot (n-i)) \tag{5.5}$$

The maximum value of the function $f(x) = nx - x^2$ in the interval $[1, n-1]$ can be determined by derivatives to be $\frac{n}{2}$. Based on this, and the fact that i is an integer, then for a line topology:

$$MNPL(G) = \begin{cases} \frac{n^2}{4} & \text{if n is even} \\ \frac{n^2-1}{4} & \text{otherwise} \end{cases} \tag{5.6}$$

The $MNPL$ for the line topology marks an upper bound on the general tree topologies. This means that another specific tree topology, which is not a line, does not have an $MNPL$ greater than the $MNPL$ of a line topology with the same number of vertices.

Star topology In a star topology represented by G, $(n-1)$ vertices in the graph are connected to a single common vertex. Based on this, $S_{(i,j)}$ is equal to $n-1$ for all the links. Therefore, for a star topology:

$$MNPL(G) = n - 1 \tag{5.7}$$

The $MNPL$ for the star topology marks a lower bound on the general tree topologies. This means that another specific tree topology, which is not a star, does not have an $MNPL$ less than the $MNPL$ of a star topology with the same number of vertices.

5.1.2 Ring topology

In a ring topology represented by G, each vertex is connected to exactly two nodes, given the set of vertices $\{v_1, v_2, ...v_i, .., v_n\}$, where:

- v_n is connected to v_1 and v_{n-1}
- v_1 its connected to v_2 and v_n
- v_i its connected to v_{i+1} and v_{i-1} where $2 \leq i \leq n-1$.

In a ring graph each pair of vertices can only be connected by exactly two different paths. Therefore, the total number of paths for any ring graph is:

$$TNP(G) = 2 \cdot n \cdot (n-1) \tag{5.8}$$

Additionally, the maximum number of paths per destination ($MNPD$) is:

$$MNPD(G) = 2 \cdot (n-1) \tag{5.9}$$

In a ring graph given a link (v_i, v_j), the number of paths that originate at v_k and use link (v_i, v_j) is equal to one, minus the number of nodes of the path going v_k to v_j without passing by v_i. Because of the characteristics of the topology, the closest node will have a value of 1 and each successive node will have one unit more until the farthest which has $n-1$. Additionally, the total values

are the same for any link. Therefore, the maximum number of paths per link ($MNPL$) in a ring topology is:

$$MNPL(G) = \sum_{i=1}^{n-1} i = \frac{n*(n-1)}{2} \tag{5.10}$$

5.1.3 Full Mesh topology

In a full mesh topology represented by G, all vertices in the graph are connected to each other, meaning that a link exists for every pair of vertices (i, j). In a full mesh topology any permutation of more than one vertex is a path; therefore, the total number of paths (TNP) is equal to:

$$TNP(G) = \sum_{i=2}^{n} \frac{n!}{(n-i)!} \tag{5.11}$$

Given that the topology is symmetrical the maximum number of paths per destination ($MNPD$) will be equal to the total number of paths divided by n, $\sum_{i=2}^{n} \frac{n!}{n(n-i)!}$. Because $n > 1$, it can be expressed as:

$$MNPD(G) = \sum_{i=2}^{n} \frac{(n-1)!}{(n-i)!} \tag{5.12}$$

To calculate the maximum number of paths per link, as the topology is completely symmetrical, the number of paths is the same on all the links. Given the set of vertices $\{v_1, v_2, ...v_i, ., ...v_j., v_n\}$, a path using the link (v_i, v_j) can be defined as any permutation of vertices that contains both v_i, and v_j, with v_j being after v_i in the sequence. Based on this, the number of paths of a link can be obtained by calculating the number of permutations of $n-2$ elements (all the vertices with the exception of v_i and v_j) multiplied by $n-1$ (for each permutation, the pair v_i,v_j could be placed in $n-1$ places) plus 1 (the path $P = \{v_i, v_j\}$ containing only the pair v_i,v_j).

It is represented by:

$$MNPL(G) = 1 + (n-1) \sum_{i=1}^{n-2} \frac{(n-2)!}{(n-2-i)!} \tag{5.13}$$

$$= 1 + \sum_{i=1}^{n-2} \frac{(n-1)!}{(n-2-i)!} \tag{5.14}$$

$$= 1 + \sum_{i=3}^{n} \frac{(n-1)!}{(n-i)!} \tag{5.15}$$

5.1.4 Topology comparison

A summary of the analytical study is presented in Table 5.1. If we compare the values of $MNPD$ and $MNPL$ for each type of topology, we would have that for:

	Line	**Ring**	**Star**	**tree**
TNP	$n*(n-1)$	$2*n*(n-1)$	$n*(n-1)$	$n*(n-1)$
$MNPL$	$\frac{n^2}{4}$ if n is even, else $\frac{n^2-1}{4}$	$\frac{n*(n-1)}{2}$	$n-1$	$\{\frac{n^2}{4}, n-1\}$
$MNPD$	$n-1$	$2*(n-1)$	$n-1$	$n-1$

	Fully connected
TNP	$\sum_{i=2}^{n} \frac{n!}{(n-i)!}$
$MNPL$	$1+\sum_{i=3}^{n} \frac{(n-1)!}{(n-i)!}$
$MNPD$	$(n-1)+\sum_{i=3}^{n} \frac{(n-1)!}{(n-i)!}$

TNP=Total number of paths, $MNPL$=maximum number of paths per link,$MNPD$=maximum number of paths per destination

Table 5.1: Maximum number of paths

- Line: $MNPL$ is higher than $MNPD$ with $MNPL-MNPD = \frac{n^2}{4}-n+1$
- Ring: $MNPL$ is higher than $MNPD$ when $n > 4$, given by $MNPL - MNPD = \frac{n^2}{2} - \frac{5*n}{2} + 2$
- Star: $MNPL$ is equal to $MNPD$
- Tree: the difference ranges from being $MNPL$ higher by $MNPL-MNPD = \frac{n^2}{4} - n + 1$ and from the two values being equal
- Full mesh: $MNPD$ is higher with $MNPD - MNPL = n - 2$.

These results show that when there is only one path between each pair of nodes (as it is the case with tree topologies), the $MNPL$ is, at most, equal to the $MNPD$. Additionally when the degree of the topology nodes is higher, the $MNPL$ is lower. Also the study demonstrates that the $MNPL$ can be higher than the $MNPD$, and vice versa depending on the topology characteristics.

5.2 Experimental study

This section describes the simulations performed to measure the effect of the topology characteristics for each type of label scope. Our simulation methodology relies on sets of topologies with specific characteristics (such as size and node degree). The IGEN [Quo05] topology generator is used to generate these topologies. All the generated topologies are evaluated in an online routing scenario. The implemented algorithm, the Shortest Path First (SPF), selects the path with the minimum cost (set to the geographical distance of the nodes). If several paths with minimum cost metric are found, the one with the minimum number of hops is selected. Only one routing metric and algorithm is evaluated because in previous sections, several algorithms with different metrics did not show a considerable difference in label space usage.

For all the topologies, the link capacity is set to 10Gb/s and bandwidth requests of 300Mb are generated. The source-destination pairs are selected randomly using a uniform distribution. For each topology, bandwidth requests are generated until no more traffic can be accommodated in the network.

Simulations are performed by analyzing both labels per link and labels per destination. Results are evaluated in terms of the number of labels used from the link or destination (depending on the label scope evaluated) with the highest amount out of all the links or destinations in the network (maximum number of used labels) and the average number of labels of all the links or destinations. Additionally, the total number of forwarding states is also analyzed.

The topology generation process consists of two steps: first, the nodes are generated and positioned on the plane; then, links are generated among these nodes. The position on the plane on the nodes affects only the geographical distance of the nodes, thus directly determine the cost of the links used for the routing algorithm.

In this section, we refer to the topology size as the number of nodes the topology has. Three different sets of topologies, each having specific characteristics, are generated. They are: the fixed size homogeneous node degree, the fixed size heterogeneous node degree and the unfixed size sets. In addition to the three generated sets, a fourth set consisting of reference topologies taken from [ea07b], is also used.

5.2.1 Fixed size homogeneous node degree set

For the fixed size homogeneous node degree set, topologies are generated using the harary heuristic. This heuristic receives as a parameter the node degree and generates a topology where all the nodes have the specified node degree. The objective of evaluating this set is to analyze how the mesh-ness of the topology affects label usage. All topologies of this set use the same set of nodes positioned on the plane; they differentiate from each other by their number of links. A set of 100 nodes, positioned randomly on the plane across a world map is used. For this demand set, 11 topologies were generated with node degree from 2 to 100 (full mesh). Even though, a 100 node full mesh topology is unrealistic, the purpose of this set is to evaluate the theoretical relationship between label space usage and topology mesh-ness. More realistic topologies are generated in the Fixed size heterogeneous node degree set.

Figure 5.1 shows the total number of forwarding states for the set. The number of states increases with the node degree. The lowest number of states was achieved by label merging, which reached up to 15% less labels than inverse trees and up to 46% less than with no technique.

Figure 5.2 shows the maximum and average number of labels for the set. For both the maximum and average number of labels, the number of labels per destination when no technique is applied increases considerably with the degree of the node (the average being seven with node degree 2 and 830 in a full mesh). When the inverse tree technique is applied, both average and maximum number of labels increase with the node degree. The average ranges from 1 to 113.5 labels and the maximum ranges from 2 to 126. When label reutilization is applied, both average and maximum number of labels still increase with the node degree. The average ranging from 2 to 19 labels and the maximum ranging from 5 to 25. For labels per link the maximum is constant with 25 labels both when applying and not applying label merging. On the other hand, the average number of labels decreases with the node degree, ranging from 24 to 16 without any technique and from 18 to 8 with label merging.

Result trends are according to expectations. The case where the number

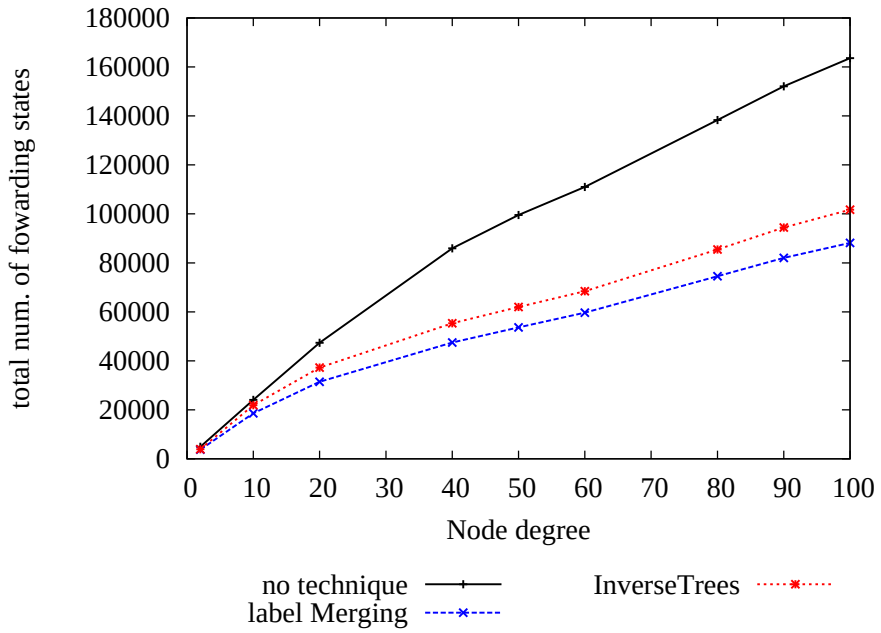


Figure 5.1: Number of forwarding states for the fixed size homogeneous node degree set

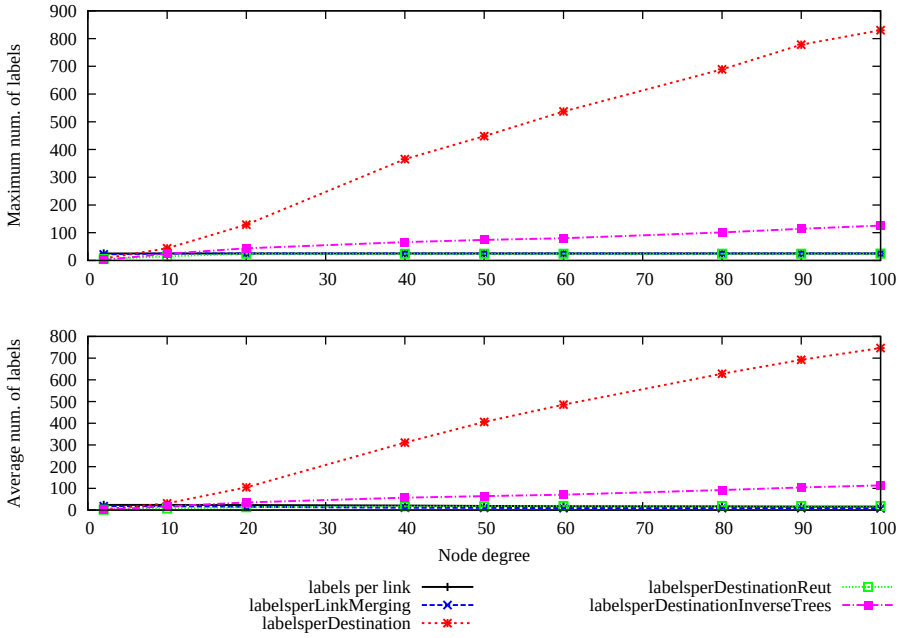


Figure 5.2: Number of labels for the Fixed size homogeneous node degree set

of labels was most affected by node degree is label per destination. This could be due to the fact that when the node degree increases the network has more links and more capacity to accommodate traffic, so more LSP are created per destination. On the other hand, in the case of labels per link scope, the number of LSPs per link does not increment because the capacity of each link is the same. When label reutilization and/or inverse trees are applied, LSP that are disjoint can use the same label, thus allowing to reduce the number of labels closer to label per link.

5.2.2 Fixed size heterogeneous node degree set

For this set, the topologies are generated using a different heuristic. Therefore, unlike the fixed size homogeneous node degree set, the nodes of the set topologies do not have the same node degree. The objective of evaluating this set is to consider a more realistic case (given that a topology where all the nodes have the same degree is unlikely) using the Waxman [Wax88] heuristic. As in the fixed size homogeneous node degree set, all the generated topologies of the set share the same set of nodes positioned on the plane, which is the same as the one used for the fixed size homogeneous node degree set. A total of 18 topologies were generated. Each of the topologies generated using the Waxman heuristic have a different value for the beta parameter. The beta parameter establishes the relationship between the probability of a link being generated and the geographical distance of the node it connects (therefore, directly affecting the node degree). The average node degrees are between 4.82 and 11.14.

Figure 5.3 shows the total number of forwarding states for the set. The behavior is similar to the previous set, when comparing the same range of node degrees.

Figure 5.4 shows the maximum and average number of labels for the set. The maximum number of labels per destination ranges from 40 to 92 without any technique and from 12 to 24 labels with inverse trees. When label reutilization is applied, the number of labels ranges from 24 to 25. The average number of labels per destination ranges from 27 to 73 without any technique, from 6 to 14 labels with inverse trees and from 12 to 14 with label reutilization. For labels per link the maximum is constant with 25 labels both when applying and not applying label merging. The average number of labels is also constant, ranging from 23 to 24 labels without any technique and from 13 to 14 labels with merging.

5.2.3 Unfixed size set

For this set, the topologies are generated using the Waxman heuristic. Each topology has a different number of nodes, and all the topologies were generated using the same parameters for the Waxman heuristic. The objective of evaluating this set is to analyze how the size of the network affects label usage. The node degree statistics of the set are illustrated in Table 5.2.

Given that the topologies in this set have different numbers of nodes, the total number of forwarding states is not comparable. Therefore, Figure 5.5 shows the average number of forwarding states per node of the set. The average number of forwarding states presents an increase proportional to the size of the network. The improvement of both techniques is constant as the size increases.

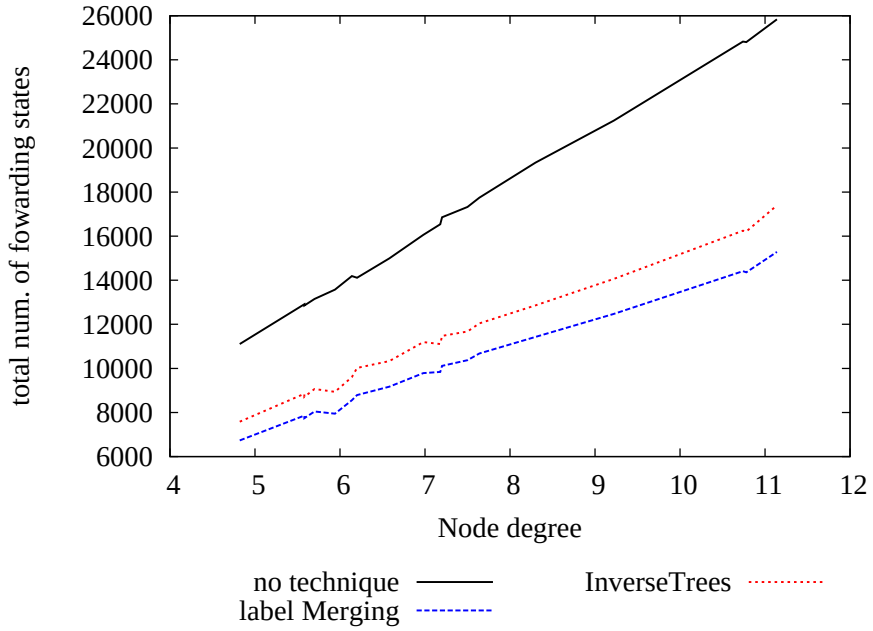


Figure 5.3: Number of forwarding states for the fixed size heterogeneous node degree set

Figure 5.6 shows the maximum and average number of labels for the set. The number of labels was more affected by the network size for labels per destination when no technique was applied. The average number of labels range from 23 to 59 and the maximum number of labels from 31 to 69. When label reutilization was applied, the average number of labels ranged from 9 to 14 and the maximum from 17 to 25. When inverse trees was applied, the average number of labels ranged from 6 to 12 and the maximum from 9 to 21.

In the case of labels per link the maximum number of labels is constant to 25 with and without merging. On the other hand, the average number of labels ranges from 17 to 23 without any technique and from 8 to 15 with merging.

5.2.4 Reference topology set

For this set, reference topologies are used. The topologies are taken from the SNDlib repository [ea07b]. The objective of evaluating the reference set is to analyze realistic topologies and to see how much their results resemble those from other sets. A total of 12 topologies are chosen from the repository; they are described in Table 5.3.

Figure 5.7 shows the average number of forwarding states for the set. For each topology of the set, the difference between techniques is always less than 5%.

Figure 5.8 shows the maximum and average number of labels for the set. When the considered techniques are applied, the differences between the average and maximum number of labels among different topologies does not exceed 15%. In cases when no techniques are applied, the difference can be up to 66%. The

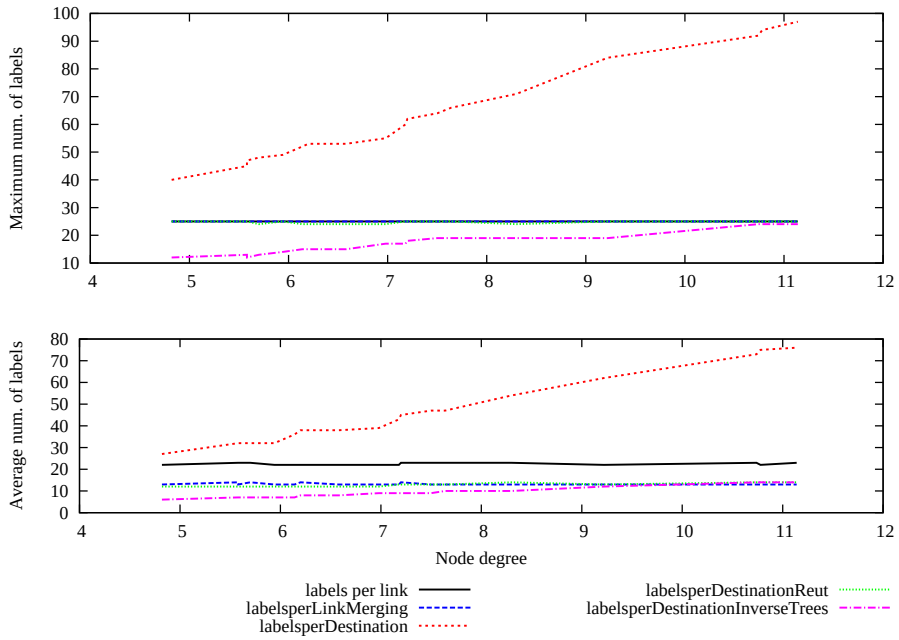


Figure 5.4: Number of labels for the Fixed size heterogeneous node degree set

size	avg	std dev	min	max
20	4,10	1,65	1	6
40	5,50	2,54	1	12
60	5,67	2,78	1	13
80	8,78	4,30	1	21
100	5,82	3,07	1	14
120	7,58	3,59	1	19
140	7,31	3,23	1	18
160	6,83	3,78	1	22
180	7,01	3,19	1	15
200	7,80	3,26	1	19

Table 5.2: Node degrees of unfixed size set

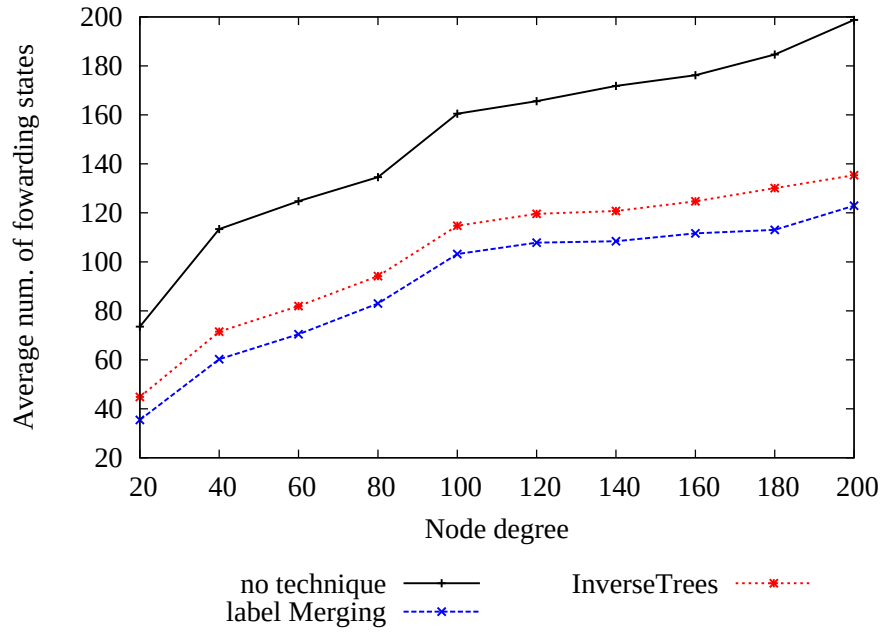


Figure 5.5: Number of forwarding states for the unfixed size set

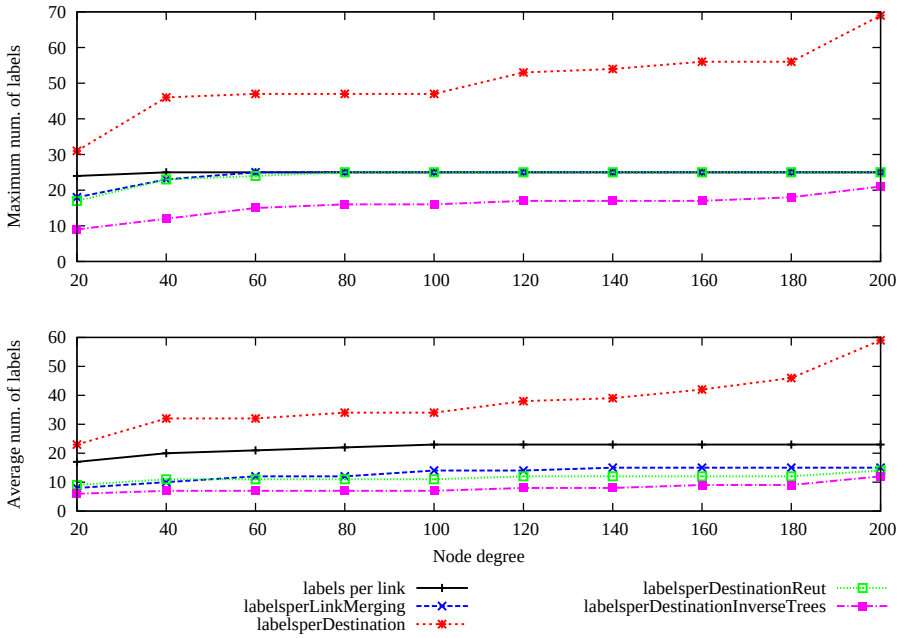


Figure 5.6: Number of labels for the unfixed size set

			Node degree		
Topology	**size**	**# links**	**avg**	**min**	**max**
atlanta	15	22	2,93	2	4
cost266	37	57	3,08	2	5
dfn-bwin	10	45	9,00	9	9
france	25	45	3,60	2	10
germany50	50	88	3,52	2	5
janos-us-ca	39	61	3,13	2	5
newyork	16	49	6,13	2	11
nobel-eu	28	41	2,93	2	5
nobel-germany	17	26	3,06	2	6
nobel-us	14	21	3,00	2	4
norway	27	51	3,78	2	6
polska	12	18	3,00	2	5

Table 5.3: Reference topology set

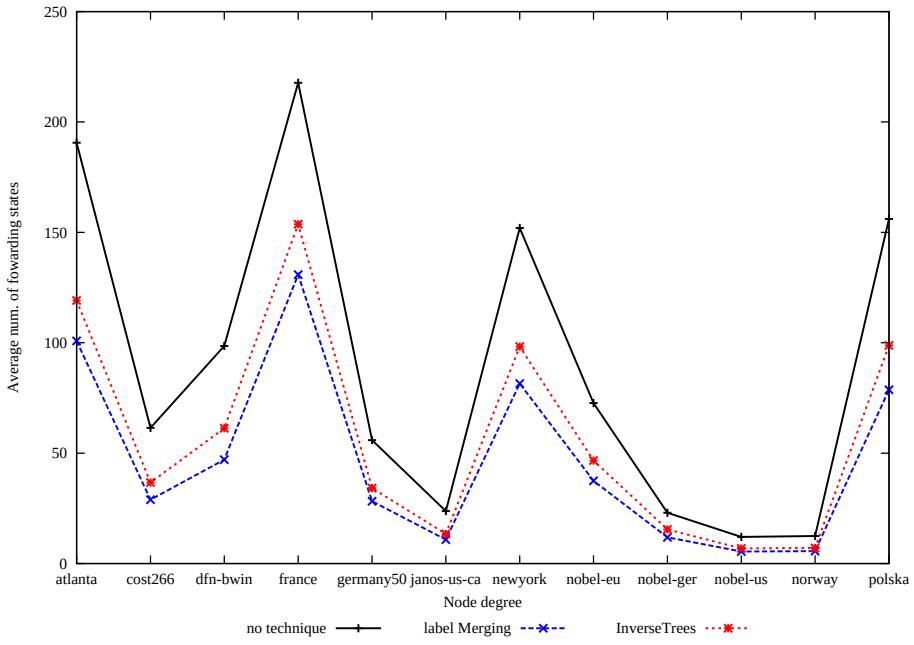


Figure 5.7: Number of forwarding states for the reference topology set

largest difference in the maximum amount of labels is for label per destination where the dfn-bwin and newyork topologies presented much higher values than the rest. Both topologies posses the highest average node degree of the set. This result corroborates the fact that the node degree has a strong impact on the number of labels used per destination.

In general, results show that even in referenced topologies the considered techniques can considerably reduce (from 66% to 15%) the impact of the topology on the number of labels.

5.3 Chapter remarks

To further conclude and complement the results obtained in Chapter 4, one of the major contributions of this thesis presented in this chapter is to study the effects of topology characteristics on carrier Ethernet label spaces. Both the number of forwarding states and the number of used labels (relevant for label exhaustion) have been considered.

The two label scopes considered are analytically studied. An upper bound on the maximum number of labels used is calculated for the basic topology types (line, star, tree, ring and full mesh). The study shows that the maximum number of labels can increase with the number of nodes for both scopes. Moreover, when there is only one path between each pair of nodes (tree results), the maximum number of labels needed with per link scope is higher than or equal to the maximum number of needed labels with per destination scope. Additionally, the study demonstrates that one label scope can use more labels than the other depending on the topology characteristics.

Simulations to evaluate the existing techniques to improve label space usage were also performed. A topology generator was used to generate topologies with specific characteristics. Three sets of topologies were generated for the experiments. Additionally a fourth one consisting of referenced topologies was considered.

Results show that the number of forwarding states increases with the size and node degrees of the topology, regardless of the technique applied. Nevertheless, when the techniques to improve label space usage are applied, considerably fewer forwarding states are needed (up to 60% fewer). Results also show that the maximum and average number of labels per destination increases considerably with the size and node degree of the topology (up to 8 labels/node degree and 0.45 labels/number of nodes). However, when the techniques considered for improving label usage are applied, both the maximum and the average number of labels do not increase considerably with the size and node degree of the network. Even when comparing different reference topologies, the maximum and average number of labels varies by, at most, 15% (without the techniques, up to 66%).

In summary, the results show that the considered techniques reduce the impact of the topology characteristics over the label space consumption when measured in terms of the number of used labels (proportional to label exhaustion). Additionally, regardless of the topology characteristics, their improvement on the number of forwarding states prevails.

Based on all these results, it can be concluded that the studied techniques for improving label space consumption are crucial to ensure the scalability of the

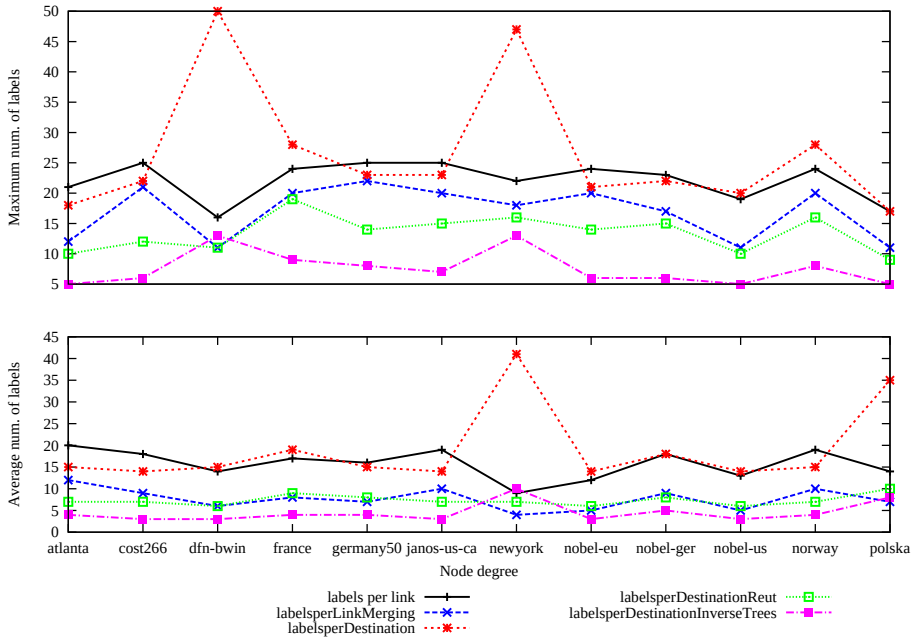


Figure 5.8: Number of labels for the reference topology set

current carrier Ethernet label-based forwarding technologies. This contribution has been published in [CPM].

Chapter 6

Performance Study of Spanning trees*

Despite all the studies that have been performed for the Spanning Tree Protocol (STP) based technologies, they are always compared either among themselves or with basic native Ethernet protocols. Additionally, to the best of our knowledge, there are no studies that can determine when label based forwarding technologies have to be used instead of STP based. Therefore, there is a need to calculate optimal performance of STP based technologies and compare them with label based forwarding ones to be able to determine, given a specific scenario, which approach to use.

In this chapter, an evaluation of the optimal performance of the STP based technologies and a comparison with label-based forwarding technologies is presented. Both offline and online routing scenarios with and without protection are considered. For the offline routing scenario, an Integer Linear Program (ILP) that calculates the optimal set of spanning trees to route a traffic matrix with or without protection, is proposed. For the online routing scenario a generalized version of the proposed algorithms (introduced in Chapter 3) is used and compared with SPF based routing algorithms. The proposed ILP can be used to determine the minimum number of trees required to optimally route all the traffic. Given a specific network and traffic matrix, the minimum number of trees can determine if STP based technologies or label based forwarding technologies have to be implemented for optimal performance.

6.1 STP-based routing generalization

In this section the scenarios and schemes, in which the STP-based technologies are evaluated and compared, are presented. We assume that any of the architectures that have been mentioned in Chapter 3 is implemented.

6.1.1 Offline routing scenario

In this section an ILP that solves the offline routing scenario using spanning trees is proposed. The objective is to evaluate the optimal performance of routing based on the fact that traffic has to follow tree routes. Three problems

are analyzed, having the same input: given a maximum number of trees $maxt$, a network graph $G = (N, E)$ and a traffic matrix $TM = N \times N$, where N is the set of nodes, and E the set of links. The problems are:

- The routing without protection problem is to find a set of undirected trees T ($|T| <= maxt$) and accommodate the traffic described by TM, such that the traffic is routed through the paths given by T. The main objective is to maximize the accommodated traffic.
- The routing with protection problem has the objective and inputs mentioned above, but the traffic matrix specifies working and backup traffic which have to be accommodated using different links.
- The minimum number of trees problem consists of accommodating all the traffic described by TM and the objective function is to minimize the number of trees used ($|T|$). In this problem TM has to be completely accommodated, this means that the matrix cannot describe traffic above the network capacity.

Unlike the ILP presented in [QMCL08], the proposed ILP does not receive T as a parameter and the model assumes that the traffic of a source-destination pair can be split through several paths. The proposed ILP is based on the multi-commodity flow problem, two models, considering routing with and without protection respectively, are presented.

Routing without protection

For each pair of nodes (s, d), where $TM(s, d) > 0$, we refer to a commodity $c \in C$ such that the requested bandwidth of the commodity $BW(c) = TM(s, d)$ and the destination and source of c are s, d, respectively.

Based on this, the proposed ILP consists of the following indices:

- i,j represent nodes in the network.
- c represents a commodity given by TM.

And the following parameters:

- BW_c set of requested bandwidth given by TM.
- $S_{(c,i)}$ is set to 1 if node i is the source of commodity c, -1 if is the destination and 0 otherwise.
- $C_{(i,j)}$ capacity of a link.
- $maxt$ maximum number of spanning trees.

The variables used in the model are the following:

- $f^{c,t}_{(i,j)}$ represents the amount of bandwidth accommodated for commodity c on link (i, j) as part of the tree t.
- $x^t_{(i,j)}$ is 1 if link (i, j) belongs to tree t, 0 otherwise.
- r^t_i is 1 if node i is the root of tree t, 0 otherwise.

- h_i^t represents the height of node i in tree t.

In order to ensure that each tree t has no cycles and is connected, the trees are modeled as unidirectional hierarchical trees, each tree has a root, and the root has height 0. If link (i, j) belongs to tree t, then $h_i^t - h_j^t = 1$, this property ensures that there are no cycles in the tree. Regardless of the fact that the trees are modeled unidirectional, the flow constraints are designed to consider them bidirectional.

The objective function is to accommodate as much bandwidth as possible through the entire network.

MAXIMIZE:

$$\sum_{j,c,t} f_{(i,j)}^{c,t} \qquad \forall i | S_{(c,i)} = 1 \tag{6.1}$$

SUBJECT TO:

- Routing constraints

$$\sum_{c,t} f_{(i,j)}^{c,t} \leq C_{(i,j)} \qquad \forall i, j \tag{6.2}$$

$$\sum_{j,t} f_{(j,i)}^{c,t} \leq BW(c) \qquad \forall i, c | S_{(c,i)} = -1 \tag{6.3}$$

$$\sum_{j,t} f_{(i,j)}^{c,t} \leq BW(c) \qquad \forall i, c | S_{(c,i)} = 1 \tag{6.4}$$

$$\sum_{j,t} f_{(j,i)}^{c,t} = 0 \qquad \forall i, c | S_{(c,i)} = 1 \tag{6.5}$$

$$\sum_{j,t} f_{(i,j)}^{c,t} = 0 \qquad \forall i, c | S_{(c,i)} = -1 \tag{6.6}$$

$$\sum_{j,t} f_{(i,j)}^{c,t} - \sum_{j,t} f_{(j,i)}^{c,t} = 0 \qquad \forall i, c | S_{(c,i)} = 0 \tag{6.7}$$

Constraint 6.2 ensures that the accommodated traffic on a link does not exceed the link capacity. Constraints 6.3 and 6.4 ensures that the accommodated traffic does not exceed the demanded traffic. Constraints 6.5,6.6 and 6.7 are the flow conservation constraints.

- Tree shape constraints

$$h_i^t \leq |N| \cdot \sum_i (x_{(i,j)}^t) \qquad \forall j,t \qquad (6.8)$$

$$h_j^t - h_i^t \geq 1 - (!x_{(i,j)}^t \cdot (|N|+1)) \qquad \forall j,i,t \qquad (6.9)$$

$$h_j^t - h_i^t \leq 1 + (!x_{(i,j)}^t \cdot |N|) \qquad \forall j,i,t \qquad (6.10)$$

$$x_{(i,j)}^t + x_{(j,i)}^t \leq 1 \qquad \forall j,i,t \qquad (6.11)$$

$$\sum_j (r_j^t) \leq 1 \qquad \forall t \qquad (6.12)$$

$$\sum_i (x_{(j,i)}^t) \geq r_j^t \qquad \forall j,t \qquad (6.13)$$

$$\sum_i (x_{(i,j)}^t) \leq !r_j^t \qquad \forall j,t \qquad (6.14)$$

$$\sum_i (x_{(j,i)}^t) \leq |N| \cdot (r_j^t + \sum_i (x_{(i,j)}^t)) \qquad \forall j,t \qquad (6.15)$$

Constraint 6.8 ensures that nodes with height zero are only nodes that have no father in the tree, which are either the root or a node not belonging to the tree. Constraints 6.9 and 6.10 ensures that the difference between the height of two connected nodes in the tree is 1. Constraint 6.11 ensures unidirectionality. Constraints 6.12 and 6.13 ensure that there is only one root per tree and that the root is connected to at least one node. Constraint 6.14 ensures that the root does not have a father, and the other nodes do not have more than one. Constraint 6.15 ensures that a node that is not the root and does not have a father, is not connected with any node.

- Tree-flow constraint

$$f_{(j,i)}^{c,t} \leq MAX_c(BW(c)) \cdot (x_{(i,j)}^t + x_{(j,i)}^t) \qquad \forall i,j,c,t \qquad (6.16)$$

The constraint ensures that the accommodated traffic follows the paths given by the trees. Note that the expression $(x_{(i,j)}^t + x_{(j,i)}^t)$ ensures that even though the trees are modeled unidirectional, traffic can flow in any direction given by the links belonging to the tree.

Routing with protection

To consider dedicated bandwidth protection an ILP model based on the previous one is presented. The protection model uses the same variables as the previous one, except $f_{(i,j)}^{c,t}$ which is removed and replaced by:

- $wf_{(i,j)}^{c,t}$ is 1 if link (i,j) is used to route the working traffic of commodity c, 0 otherwise.
- $bf_{(i,j)}^{c,t}$ is 1 if link (i,j) is used to route the backup traffic of commodity c, 0 otherwise.

To be able to support protection the bandwidth of a commodity is no longer splittable (as flow assignment is binary instead of continuous as in the previous model). However in order to support traffic splitting then several commodities can be specified per each source destination pair.

- To support protection, all the routing constraints are replaced by:

$$BW(c) \cdot \sum_{c,t} (wf^{c,t}_{(i,j)} + bf^{c,t}_{(i,j)}) \leq C_{(i,j)} \qquad \forall i,j \qquad (6.17)$$

$$\sum_{j,t} wf^{c,t}_{(i,j)} - \sum_{j,t} wf^{c,t}_{(j,i)} = 0 \qquad \forall i,c|S_{(c,i)} = 0 \qquad (6.18)$$

$$\sum_{j,t} bf^{c,t}_{(i,j)} - \sum_{j,t} bf^{c,t}_{(j,i)} = 0 \qquad \forall i,c|S_{(c,i)} = 0 \qquad (6.19)$$

$$\sum_{j,t} wf^{c,t}_{(j,i)} \leq 1 \qquad \sum_{j,t} bf^{c,t}_{(j,i)} \leq 1 \qquad \forall i,c|S_{(c,i)} = -1 \qquad (6.20)$$

$$\sum_{j,t} wf^{c,t}_{(i,j)} \leq 1 \qquad \sum_{j,t} bf^{c,t}_{(i,j)} \leq 1 \qquad \forall i,c|S_{(c,i)} = 1 \qquad (6.21)$$

$$\sum_{j,t} wf^{c,t}_{(j,i)} = 0 \qquad \sum_{j,t} bf^{c,t}_{(j,i)} = 0 \qquad \forall i,c|S_{(c,i)} = 1 \qquad (6.22)$$

$$\sum_{j,t} wf^{c,t}_{(i,j)} = 0 \qquad \sum_{j,t} bf^{c,t}_{(i,j)} = 0 \qquad \forall i,c|S_{(c,i)} = -1 \qquad (6.23)$$

- The tree-flow constraint is also replaced by:

$$wf^{c,t}_{(j,i)} \leq x^t_{(i,j)} + x^t_{(j,i)} \qquad \forall i,j,c,t \qquad (6.24)$$

$$bf^{c,t}_{(j,i)} \leq x^t_{(i,j)} + x^t_{(j,i)} \qquad \forall i,j,c,t \qquad (6.25)$$

All these constraints have the same function as the ones of the previous model. Additionally, the following protection constraints are also added:

$$\sum_{t} (wf^{c,t}_{(j,i)}) + \sum_{t} (bf^{c,t}_{(j,i)}) \leq 1 \qquad \forall i,j,c \qquad (6.26)$$

$$\sum_{j,t} wf^{c,t}_{(i,j)} - \sum_{j,t} bf^{c,t}_{(i,j)} = 0 \qquad \forall i,c|S_{(c,i)} = 1 \qquad (6.27)$$

Constraint 6.26 guarantees backup/working traffic disjointness and constraint 6.27 ensures that all traffic is protected. All the tree shape constraints are the same as in the previous model. Finally the objective function is replaced by:

MAXIMIZE:

$$BW(c) \cdot \sum_{j,c,t} wf^{c,t}_{(i,j)} \qquad \forall i|S_{(c,i)} = 1 \qquad (6.28)$$

Minimum Number of Trees Model

In order to calculate the minimum number of trees for the models proposed above, the objective function and some constraints need to be replaced. Given that we want to calculate the minimum number of trees needed to accommodate all the traffic matrix, routing constraints have to force all traffic to be routed and the objective function set to minimize the number of trees.

For the model without protection, constraints 6.3 and 6.4 have to be replaced by:

$$\sum_{j,t} f^{c,t}_{(j,i)} = BW(c) \qquad \forall i, c | S_{(c,i)} = -1 \tag{6.29}$$

$$\sum_{j,t} f^{c,t}_{(i,j)} = BW(c) \qquad \forall i, c | S_{(c,i)} = 1 \tag{6.30}$$

For the model with protection, constraints 6.20 and 6.21 have to be replaced by:

$$\sum_{j,t} w f^{c,t}_{(j,i)} = 1 \qquad \sum_{j,t} b f^{c,t}_{(j,i)} = 1 \qquad \forall i, c | S_{(c,i)} = -1 \tag{6.31}$$

$$\sum_{j,t} w f^{c,t}_{(i,j)} = 1 \qquad \sum_{j,t} b f^{c,t}_{(i,j)} = 1 \qquad \forall i, c | S_{(c,i)} = 1 \tag{6.32}$$

For both with and without protection models, the objective function must be set to:

MINIMIZE:

$$\sum_{i,t} r^t_i \qquad \forall i \tag{6.33}$$

This model is proposed as a network planning tool to determine when to use STP-based technologies or label-based forwarding techniques. For a specific network, the model can be applied for several traffic matrices and if the minimum number of trees is considerably higher than the number of trees the network switches support, then label-based forwarding techniques must be implemented for optimal performance.

6.1.2 Online routing scenario

In this section the use of STP-based technologies under the online routing scenario is explained. In the online scenario, given the topology of the network, an already accommodated traffic and a new incoming bandwidth request between two nodes, the problem is to find a path across the network that satisfies the bandwidth request. The objective is to decrease the probability of future bandwidth requests being blocked. Unlike the offline scenario, the traffic matrix is unknown and bandwidth requests arrive sequentially. In the case of STP-based technologies, a set of established spanning trees T and a maximum number of trees $maxt$ is also given. It is assumed that the spanning trees are calculated

dynamically as the routing algorithm determines it necessary. Therefore in addition to finding a path that accommodates the bandwidth request the algorithm must also modify T (either by adding trees or adding links to a existing tree) if necessary to be able to accommodate the bandwidth request.

Routing in this scenario is performed based on the link cost and using the path aggregation algorithm proposed in [SGNC04]. Given a bandwidth request between nodes (s, d), the routing algorithm first tries to see if any of the existing trees in T can be used (using the path aggregation algorithm). If none of the trees in T can be used then it creates a new tree if $|T| < maxt$, otherwise the request is rejected.

6.2 Experimental Results

In this section the performance of STP based technologies in comparison with the label-based forwarding techniques is evaluated.

Two types of topologies have been used in the related work presented in Section III: grid topologies (for example in [QMCL08]) and defined topologies (for example in [INB+07]). Similarly, two topologies are considered in this section: a grid topology of 36 (6 x 6) nodes and the defined cost266 topology [IKM03]. For both topologies link capacity is set to 10Gb/s. For all the experiments it is assumed that all nodes are sources and destinations, i.e. traffic is generated among all nodes.

6.2.1 Offline scenario

For the offline scenario, the proposed ILP models are implemented to determine the optimal performance of the STP-based technologies. For modeling the label based forwarding technologies the proposed models are modified. In the case of the model without protection, $f^{c,t}_{(i,j)}$ is replaced by $f^{c}_{(i,j)}$, and the rest of the variables are removed. Additionally all the tree shape and tree-flow constraints are removed. In the case of the model with protection, $wf^{c,t}_{(i,j)}$ and $bf^{c,t}_{(i,j)}$ are replaced by $wf^{c}_{(i,j)}$ and $bf^{c}_{(i,j)}$. As in the previous model the rest of the variables are removed together with all the tree shape and tree-flow constraints. In order to perform a fair comparison, the objective functions are the same but using the replaced variables. The traffic between source and destination is uniformly distributed between [100,1024]Mb/s. The models are solved using the Xpress-Optimizer [Ass04].

Performance is evaluated in terms of the accommodated traffic and the total reserved capacity. The accommodated traffic is the sum of the amount of traffic that is routed through all the sources and destinations, it is the objective function of the proposed models. The total reserved capacity is the sum of the capacity reserved for protection purposes in each link of the network. These values are measured versus the number of allowed trees ($maxt$) parameter specified in the model. The minimum number of trees model (MNTM) is used to calculate the minimum number of trees needed to accommodate the same amount of traffic accommodated by label-based forwarding technologies. It is represented as a vertical dotted line. The results for label-based forwarding technologies, given that they are not subject to the maximum number of trees $maxt$, are plotted as a constant horizontal line among the number of trees. This

means that only one value is calculated for the label-based forwarding per plot. On the other hand, for the STP-based, one value (represented as a point in the line) for each of the different number of trees ($maxt$) is calculated.

Results of the model without protection are presented in Figure 6.1. Results show that when using just one tree the optimal performance of the STP-based technologies is between 36% (grid) and 41% (cost266) less than the label-based forwarding ones. The minimum number of trees that give the same performance as label-based technologies is 70 and 110 for the cost266 and grid topologies, respectively. If the total accommodated bandwidth is divided by the number of trees, then the average accommodated traffic per tree is 4381Mb/s (for grid) and 5406Mb/s (for cost266). This means that even though in the grid topology more traffic can be routed, in the cost266 topology more traffic can be routed per tree.

Results of the model supporting protection are presented in Figure 6.2 and 6.3. Figure 6.2 shows the accommodated traffic. The behavior is similar to the model without protection, but the accommodated traffic is considerable less as protection capacity needs also to be reserved in this model. Additionally the minimum number of trees that gives the same performance as label-based technologies is 90 and 120 for cost and grid topologies, respectively. The average accommodated traffic per tree is 2007Mb/s (for grid) and 2055 Mb/s (for cost266). Both topologies presented similar total reserved capacity. It is important to note that given that the model was not optimizing this metric, it is possible that STP-based technologies present higher or lower values even with a high number of trees. However, when the STP-based technologies had the same performance as label-based ones in terms of accommodated traffic, the STP-based technologies reserved around 2% more bandwidth. It is not a considerable amount but it represents a drawback to using STP-based technologies.

6.2.2 Online scenario

For the online scenario, the routing scheme specified in Section 6.1.2 is used for the STP-based technologies. For comparing them with the label based forwarding technologies a Constraint Shortest Path First (CSPF) algorithm is used. The STP-based routing scheme calculates the spanning trees based on minimum hop count trees (as all the links have been assigned the same cost). The CSPF selects the path with the minimum hop count. If several paths with the minimum hop count are found, the one with the maximum residual capacity is selected. For calculating the working and backup paths, the CSPF selects the working path first, prunes the path links and then selects the backup path. Both paths need to be selected for the bandwidth request to be accepted and accommodated.

Bandwidth requests are generated between [100,200,300] Mb/s following a uniform distribution. Routes are computed sequentially according to the generated bandwidth requests. For each topology, bandwidth requests are generated until no more traffic can be accommodated in the network. Each result is the average of 10 rounds of simulation run. As in the offline scenario, performance is evaluated in terms of the accommodated traffic and the total reserved capacity. Results are also plotted versus the maximum number of allowed trees in the network. As in the offline scenario, only one value is calculated for the label-based forwarding per plot and is represented by a constant line.

Results are presented in Figures 6.4 and 6.5. Figure 6.4 shows the accommodated traffic. When 2 trees are used the performance of the STP-based technologies is between 43% and 67% less than the label-based forwarding ones, which means their difference is higher than in the offline scenario. Nevertheless when the number of trees increases the performance rises. Finally, when the number of 100 trees is reached, the performance stabilizes and does not increase further. With 100 trees the performance is still 10% and 9% less than the label-based forwarding technologies.

Figure 6.5 shows the total reserved capacity. The curves are similar to the accommodated traffic, however the reserved capacity of the STP-based technologies did not exceed that of the label-based forwarding techniques. When the number of 100 trees is reached the reserved bandwidth for the cost266 network is 8% less than the one of label-based forwarding techniques. However, in the same case for the grid network the reserved bandwidth is 1% less.

6.2.3 Common results

In addition to the accommodated traffic and the total reserved capacity, the average hop count of the used paths per bandwidth request is also evaluated. The results are very similar for both offline and online scenarios. Results for the cost266 topology showed that the average hop count for the label-based forwarding technologies was around 3 and for the STP-based, it was always higher, ranging between 3.2 and 4 (up to 25% higher). In the case of the grid topology, the average hop count for the label-based forwarding technologies was around 3.2 and for the STP-based was also higher ranging between 4 and 4.3 (also up to 25% higher). This result has to be taken into account when considering the use of STP-based technologies in cases where metrics that are affected by the length of the paths are being optimized.

6.3 Chapter remarks

This Chapter presents one of the major contributions of this thesis by studying the performance of Carrier Ethernet schemes protection. The optimal performance in resource allocation when using spanning trees for both supporting protection and/or path diversity is evaluated using an ILP. Additionally one of the existing heuristics is also evaluated, and compared with label-based forwarding technologies.

The proposed ILP models a network with or without protection mechanism. The model, given the number of allowed trees $maxt$, calculates how to accommodate the maximum amount of traffic and set the trees to support it. It can additionally calculate the minimum number of trees required to accommodate all the traffic.

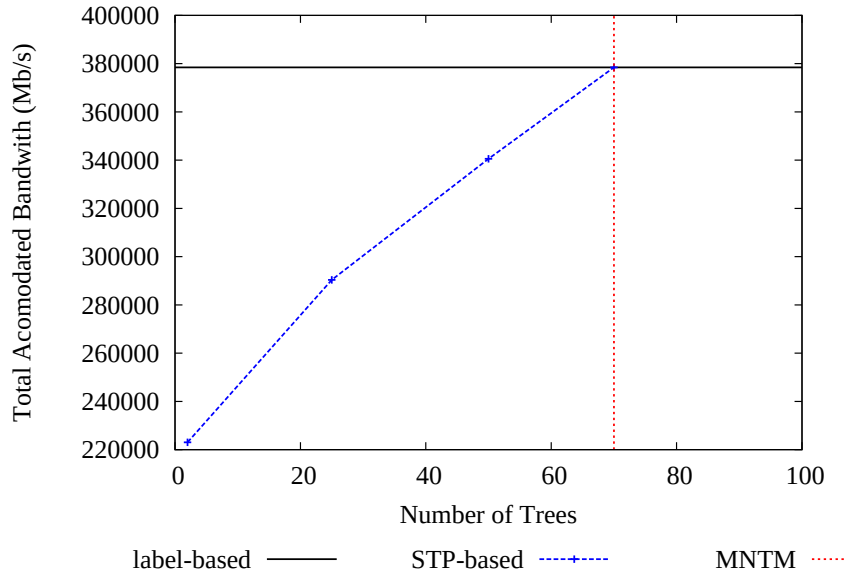
In summary, our experimental results show that an optimal use of multiple spanning trees can make the STP-based technologies accommodate the same amount of traffic as the label-based forwarding ones. In the case of protection scenarios, the STP-based technologies require a little more of reserved bandwidth (2%) to protect the same amount of traffic as label-based ones. Results also show for the implemented topologies and bandwidth demands, the minimum number of trees that needs to be supported by the network in order

to obtain optimal traffic allocation. When considering protection scenario the minimum number of required trees is between 9% and 28% more than when protection is not considered. This is one of the most important results as it serves as a guideline for network administrators when evaluating which carrier Ethernet technology to implement in a particular study case.

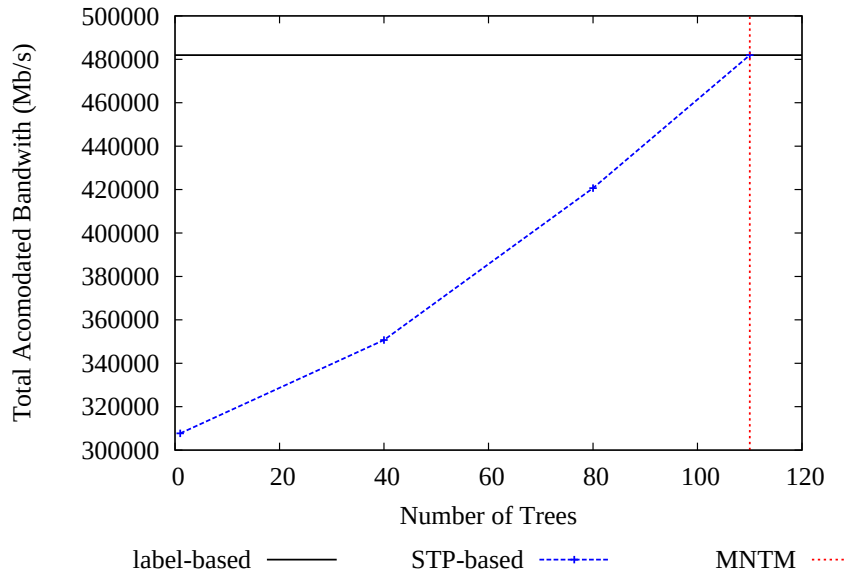
Results also give an overview of how far the performance of the existing online heuristic is in comparison with the optimal given by the ILP proposed in this document. In the studied scenario, the heuristic reaches 90% of the maximum accommodated bandwidth obtained with the optimal solution. This means that the related work improvement on the STP-based technologies allocation capabilities has been considerable even when compared against label-based approaches.

For all the evaluated scenarios, STP-based approaches show a tendency to find longer paths even when using the optimal number of trees. This result reflects that label-based forwarding has better performance when evaluating metrics affected by the length of the paths. It was also observed that the performance of the evaluated technologies can vary considerably between two topologies of different characteristics (grid and defined topologies in the evaluated case).

Finally, the proposed ILP can be used to determine the number of trees the network must support for allowing STP based technologies have an optimal performance. This can be taken into consideration in network planning to decide if label-based forwarding technologies are needed. Part of this contribution has been published in [CPM08b, CPM09b].

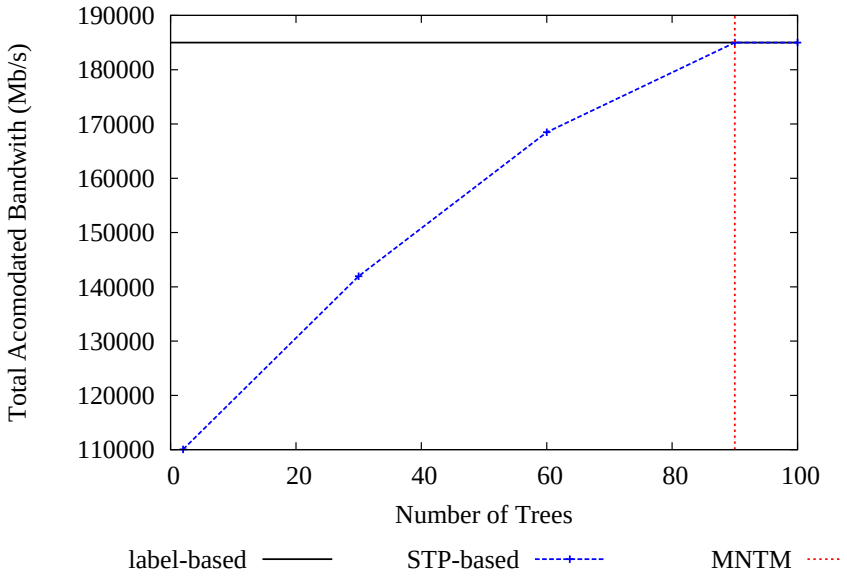


(a) Cost266

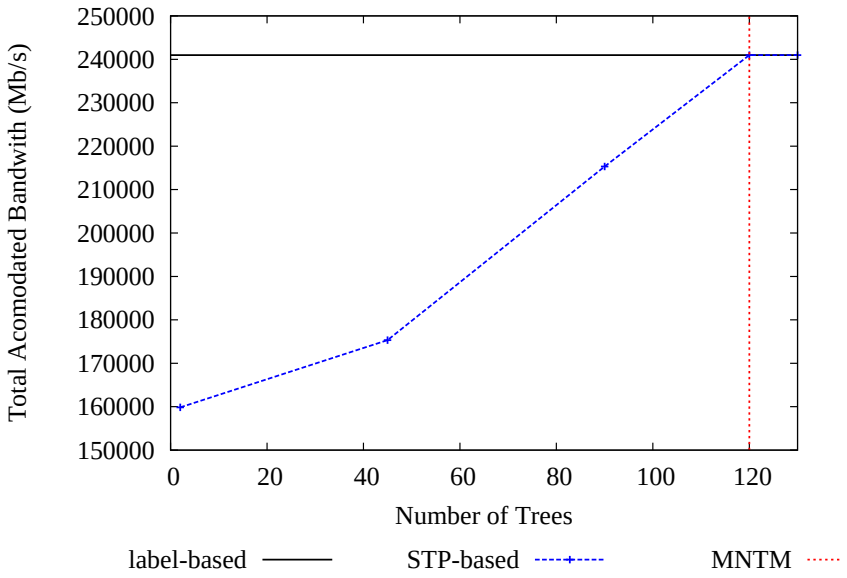


(b) Grid

Figure 6.1: Traffic accommodated for no protection model

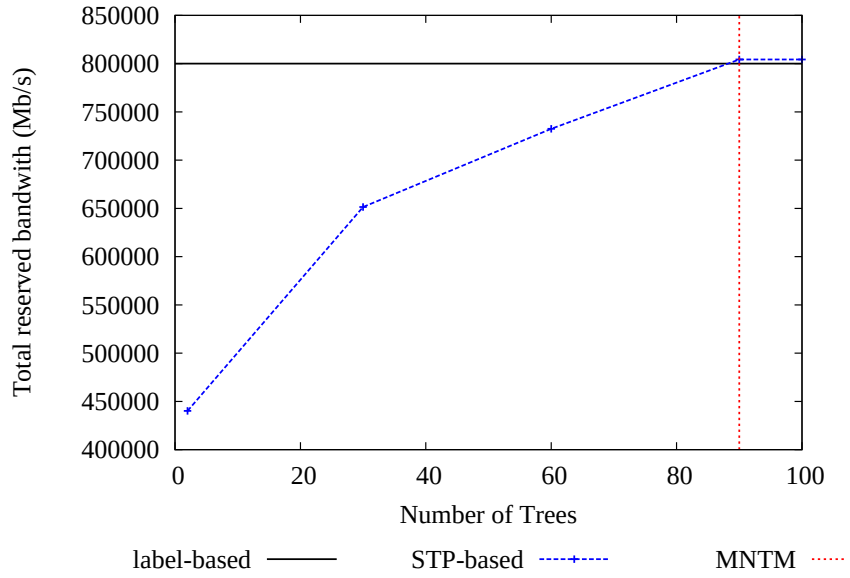


(a) Cost266

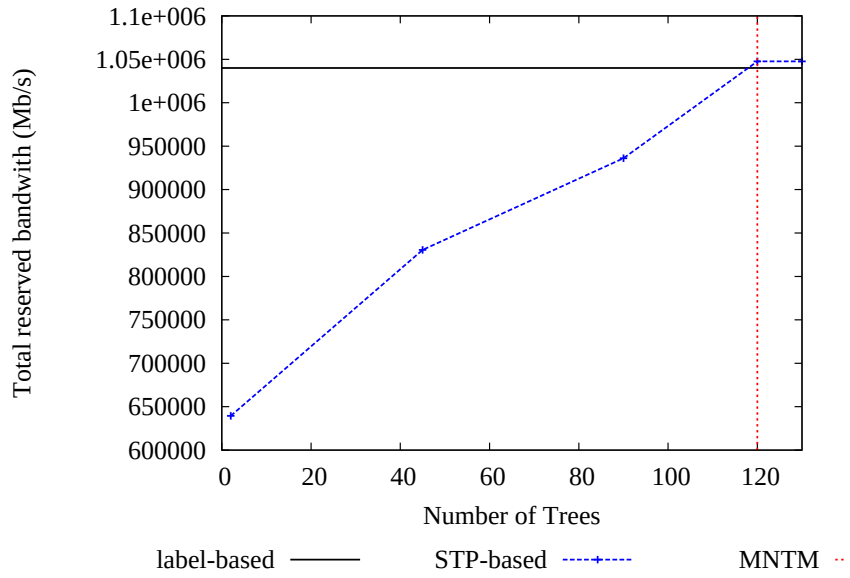


(b) Grid

Figure 6.2: Traffic accommodated for protection model

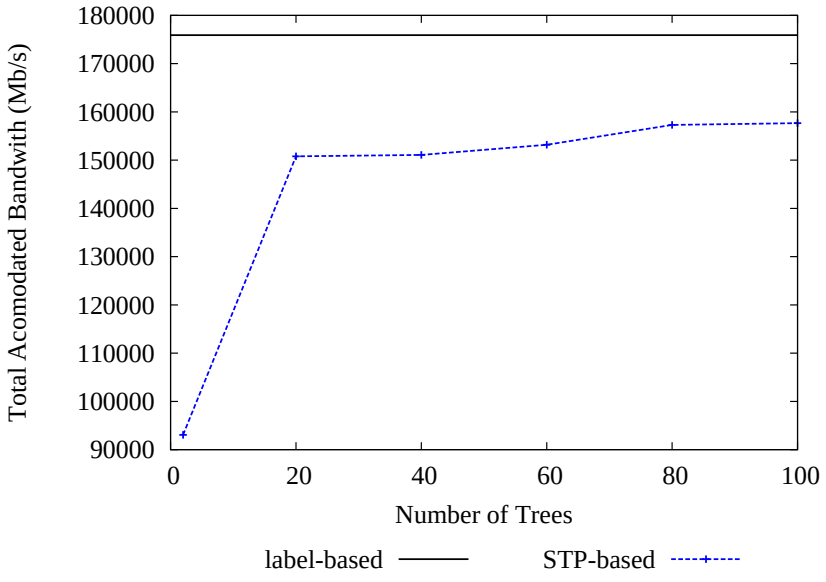


(a) Cost266

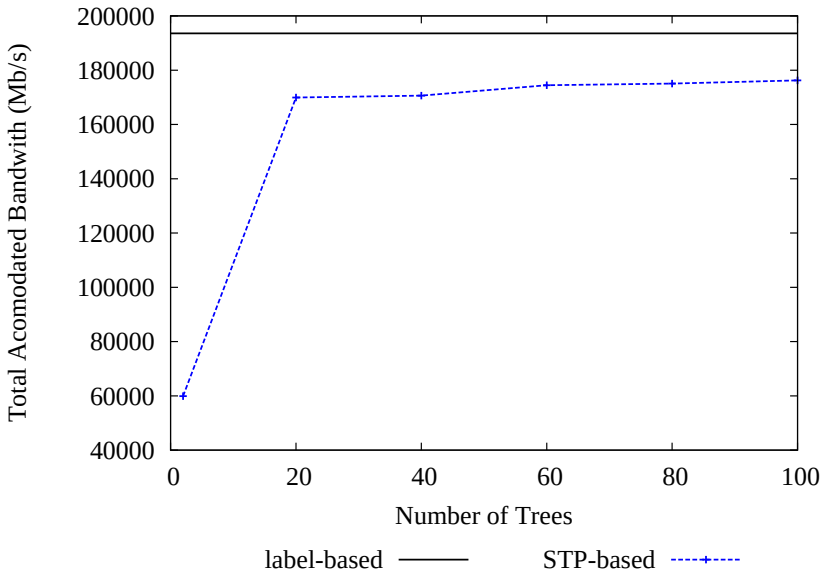


(b) Grid

Figure 6.3: Total reserved capacity for protection model

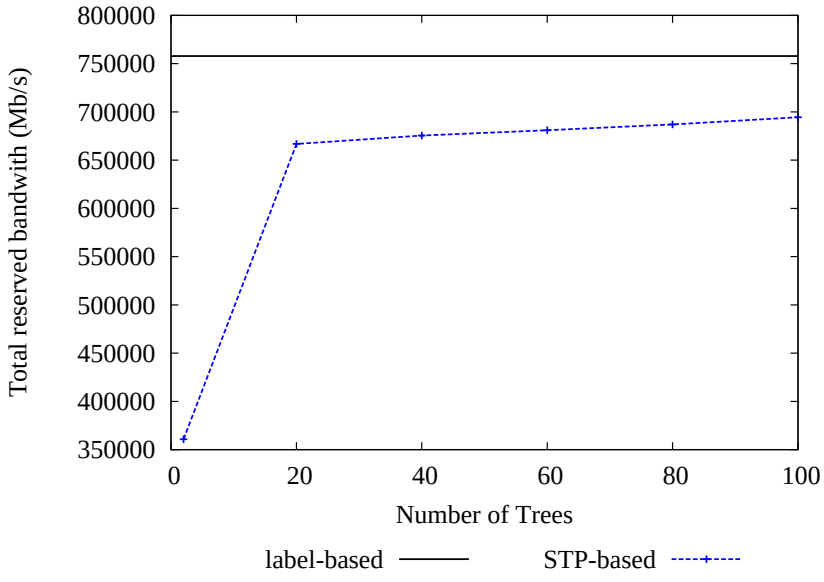


(a) Cost266

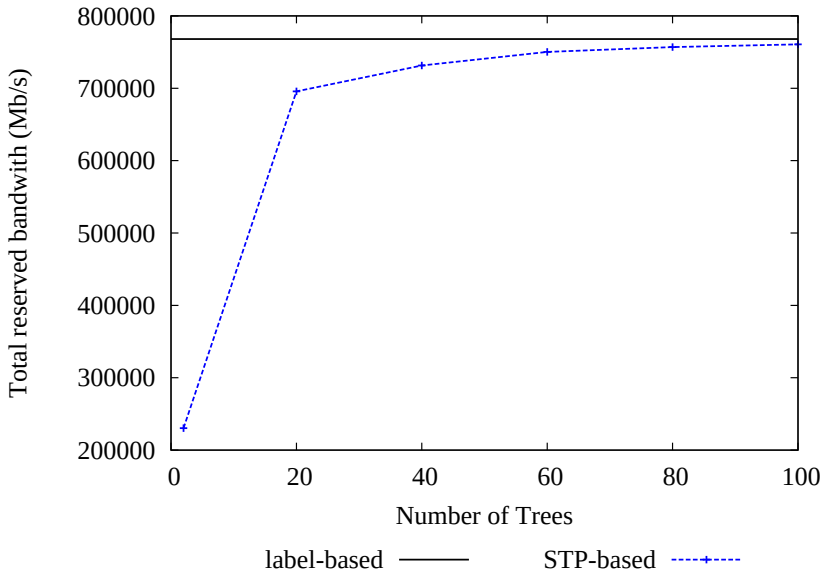


(b) Grid

Figure 6.4: Traffic accommodated for online scenario



(a) Cost266



(b) Grid

Figure 6.5: Total reserved capacity for online scenario

CHAPTER 6. PERFORMANCE STUDY OF SPANNING TREES*

Chapter 7

Conclusions and future work

7.1 Thesis conclusions

In this thesis, the scalability of different carrier Ethernet technologies is analyzed and studied. Two types of carrier Ethernet technologies are considered: Label based forwarding and STP-based technologies. Two label based forwarding technologies are considered: Ethernet VLAN-Label Switching (ELS) and Provider Backbone Bridges - Traffic Engineering (PBB-TE). The bandwidth granularity associated to labels is analyzed as an indicator of the possibility of sparsity of labels. Several available techniques that can be used to improve label scalability are reviewed and analyzed. Additionally, for ELS, a new online routing algorithm designed to take advantage of label merging is proposed and evaluated as well. For PBB-TE the VLAN-reutilization technique is formalized and the complexity of optimally applying it analyzed. A set of simulations are performed for three reference topologies. Results for the offline routing scenario show that even without applying any technique both technologies do not present label limitations. Results for the online routing scenario for both technologies show that for demands of low granularity performance degradation can be presented when no label reduction techniques are used. However applying the evaluated and proposed techniques allows performance to be maintained in terms of accommodated traffic load.

To further conclude and complement these results, the effects of topology characteristics on carrier Ethernet label spaces are also studied. The study considered both the number of forwarding states and the number of used labels (relevant for label exhaustion). Results show that the number of forwarding states increases with the size and node degrees of the topology, regardless of the technique applied. Nevertheless, when the techniques to improve label space usage are applied, considerably fewer forwarding states in the nodes are needed. Results also show that the considered techniques reduce the impact of the topology characteristics over the label space consumption when measured in terms of the number of used labels (proportional to label exhaustion).

In addition to studying label space usage, the performance of STP based technologies against label based ones is also studied. For this purpose the opti-

mal performance in resource allocation when STP based technologies for both supporting protection and/or path diversity is evaluated using an ILP. The ILP models a network with or without protection mechanism. The model calculates how to accommodate the maximum amount of traffic and set the trees to support it. It can additionally calculate the minimum number of trees required to accommodate all the traffic. Additionally one of the existing heuristics is also evaluated, and compared with label-based forwarding technologies.

Experimental results show that an optimal use of multiple spanning trees can make the STP-based technologies accommodate the same amount of traffic as the label-based forwarding ones. In the case of protection scenarios, the STP-based technologies require more reserved bandwidth (about 2%) to protect the same amount of traffic than label-based ones.

Results also give an overview of how far the performance of the existing online heuristic is in comparison with the optimal given by the ILP proposed in this document. In the studied scenario, the heuristic reaches 90% of the maximum accommodated bandwidth obtained with the optimal solution. This means that the related work improvement on the STP-based technologies allocation capabilities has been considerable even when versus label-based approaches.

Based on all these results it can be concluded that:

- The studied techniques for improving label space consumption are crucial to ensure the scalability of the current carrier Ethernet label-based forwarding technologies.

- STP-based approaches showed a tendency to find longer paths even when using the optimal number of trees, reflecting that label-based forwarding has better performance when evaluating metrics affected by the length of the paths.

- The proposed ILP can be used to determine the number of trees the network must support to allow STP based technologies to have optimal performance. This can be taken into consideration in network planning to decide if label-based forwarding technologies are needed.

7.2 Future work

There are several topics in which the present research work can be extended in the future.

7.2.1 Evaluation of protection and recovery times

Chapter 6 compared carrier Ethernet technologies protection capabilities. The technologies were evaluated in terms of the reserved capacity for ensuring protection. Nevertheless, to also evaluate restoration and protection time is crucial to further determine the performance of the technologies. This includes the comparison of the methods given by GMPLS for label based forwarding against the different recovery methods proposed for each STP based implementation.

7.2.2 Evaluation of shared protection

Shared protection was not considered in Chapter 6 because it was outside of the scope of this document. Nevertheless, there is a need to investigate if it is possible to offer shared protection on STP-based technologies and compare its performance against label based technologies.

7.2.3 Creating trees by column generation

The simulations of the proposed ILP model over the considered topologies had acceptable running time for daily network management. However, for topologies with higher number of nodes, the model must be upgraded. One of the characteristics that increases the model complexity is that the set of spanning trees is not given. Based on this, the complexity of the model can be reduced by generating the set of spanning trees using column generation.

7.2.4 Introduction of carrier Ethernet in IP/WDM networks

Advances in optical WDM networks aim to have an architecture where the IP protocol works directly over a Wavelength Switched Optical Networks (WSON) layer. On the other hand, Ethernet has positioned itself as the transport technology of choice. There is a need to study the rationale behind the Ethernet transport technologies role in the IP over WSON model. The role of electric switching in optical network has been studied in traffic grooming (e.g. [SCdO$^+$07]), however, the specific advantages of implementing Ethernet as a grooming technology must be evaluated. The study includes an analysis of the traffic conditions that would justify an IP traffic off loading over Ethernet technologies in different sections of the network (access, core...etc).

7.2.5 Multi-domain scenario

This document studied the limitations of carrier Ethernet technologies in single-domain scenarios. It is also necessary to evaluate the performance, limitations and compatibility of carrier Ethernet technologies in a multi-domain scenario. In the case of label-based forwarding technologies, this would include studying and comparing if the per destination label scope presents any issues in this scenario.

Appendix A

Author publications

This thesis has been elaborated as part of Luis Fernando Caro Perez PhD studies. During his PhD Luis has done research on two main topics, Traffic grooming in optical networks and improving carrier Ethernet technologies (covered fully by this thesis).

Traffic grooming in optical networks

Even though it is not strictly related to this thesis, research in this area helped the author gain experience in network simulation and routing ILP models.

Journals

Fernando Solano, Luis Caro, Jaudelice de Oliveira, Ramon Fabregat, and Jose Marzo. G+: Enhanced traffic grooming in WDM mesh networks using Lighttours. IEEE Journal on Selected Areas in Communications (JSAC), June 2007.

Conferences

Javier E. Sierra, Luis F. Caro, Fernando Solano, Jose L. Marzo, Ramon Fabregat and Yezid Donoso. All-optical Unicast/Multicast Routing in WDM Networks. Published at GLOBECOM 2008.

Sierra, Luis F. Caro, Fernando Solano, Jose L. Marzo, Ramon Fabregat and Yezid Donoso. Dynamic Unicast/Multicast Traffic Grooming Using S/G Lighttree in WDM Networks. Javier E. Published at SPECTS. June 2008.

Javier Sierra, Luis F. Caro, Fernando Solano, Ramon Fabregat, Yezid Donoso. S/G Light-tree: Multicast Grooming Architecture for Improved Resource Allocation. Published at IEEE, VII Workshop in G/MPLS networks. March 2008.

J L. Marzo, L F. Caro, F Solano, J. de Oliveira, R Fabregat. Operational Cost Reduction in WDM Networks using Lighttours (invited). Published at International Conference on Transparent Optical Networks (ICTON) proceedings. July 2007.

J.L. Marzo, F. Solano, J.C. de Oliveira, L.F. Caro, R. Fabregat. Optimal traffic grooming in WDM using lighttours (invited). Published at International Conference on Transparent Optical Networks ICTON. June 2006.

Fernando Solano, L F. Caro , Ramon Fabregat, Jose Luis Marzo, T. K. Stidsen. Enhancing Traffic Grooming in WDM Networks through lambda-monitoring. Published at Eighth INFORMS Telecommunications Conference. April 2006.

Improving carrier Ethernet technologies

Publications in this topic cover all the contributions of this thesis.

Journals

Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. Ethernet label spaces dependency on network topology. Accepted in European Transactions on Telecommunications.

Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. Enhancing label space usage for Ethernet VLAN-label switching. Computer Networks. 2009.

Conferences

Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. Comparison of MSTP and gels performance. In Benchmarking Carrier Ethernet Technologies workshop collocated with NGI 2008, April 2008.

Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. Improving label space usage for Ethernet label switched paths. In Proc. IEEE International Conference on Communications (ICC 2008), May 2008.

Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. Carrier Ethernet label scalability. In Proc. 12th International Telecommunications Network Strategy and Planning Symposium. NETWORKS 2008., September 2008.

Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. A performance analysis of carrier Ethernet schemes based on multiple spanning trees. In Proc. VIII Workshop in G/MPLS networks, June 2009.

Bibliography

[802a] IEEE 802.1Qay. *IEEE Standard for Local and Metropolitan Area Networks—Virtual Bridged Local Area Networks - Amendment: Provider Backbone Bridge Traffic Engineering.*

[802b] IEEE 802.1s. *IEEE Standard for Local and Metropolitan Area Networks, Multiple Spanning Trees.* IEEE.

[80203a] IEEE 802.1D. *IEEE Standard for local and metropolitan area networks: Media Access Control (MAC) Bridges.* IEEE, 2003.

[80203b] IEEE 802.1Q. *IEEE standard for local and Metropolitan Area Networks: Virtual Bridged Local Area Networks.* IEEE, 2003.

[80204] IEEE 802.1w. *IEEE Standard for local and metropolitan area networks: Rapid Reconfiguration of Spanning Tree.* IEEE, 2004.

[80205] IEEE 802.1ad. *IEEE Standard for local and metropolitan area networks: Provider Bridges.* IEEE, 2005.

[80208] IEEE 802.1ah. *IEEE Standard for local and metropolitan area networks: Provider Backbone Bridges.* IEEE, 2008.

[AA05] M.C. Ali and G.A.G. Alcatel. Traffic engineering in metro ethernet. *Network, IEEE*, 19(2):10–17, 2005.

[Ass04] D. Associates. Xpress-Mosel Reference Manuals and Xpress-Optimizer Reference Manual. *Release 2004G*, 2004.

[AT03] D. Applegate and M. Thorup. Load optimal MPLS routing with N+ M labels. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 2003.

[BGN03] S. Bhatnagar, S. Ganguly, and B. Nath. Creating multipoint-to-point LSPs for traffic engineering. *High Performance Switching and Routing, 2003, HPSR. Workshop on*, pages 201–207, 2003.

[CGK92] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: an approach to high bandwidth optical wan's. *Communications, IEEE Transactions on*, 40(7):1171–1182, Jul 1992.

[CPM] Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. Ethernet label spaces dependency on network topology. *Accepted in European Transactions on Telecommunications.*

[CPM08a] Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. Carrier ethernet label scalability. In *Proc. 12th International Telecommunications Network Strategy and Planning Symposium. NETWORKS 2008.*, September 2008.

[CPM08b] Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. Comparison of mstp and gels performance. In *Benchmarking Carrier Ethernet Technologies workshop co-located with NGI 2008*, April 2008.

[CPM08c] Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. Improving label space usage for ethernet label switched paths. In *Proc. IEEE International Conference on Communications (ICC 2008)*, May 2008.

[CPM09a] Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. Enhancing label space usage for ethernet vlan-label switching. *Computer Networks*, 53(7):1050–1061, 2009.

[CPM09b] Luis F. Caro, Dimitri Papadimitriou, and Jose L. Marzo. A performance analysis of carrier ethernet schemes based on multiple spanning trees. In *Proc. VIII Workshop in G/MPLS networks*, June 2009.

[DS06] Amaro F. De Sousa. Improving load balance and resilience of ethernet carrier networks with ieee 802.1s multiple spanning tree protocol. In *ICNICONSMCL '06: Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, page 95, Washington, DC, USA, 2006. IEEE Computer Society.

[ea07a] Lauren Ciavaglia et al. Tiger: Optimizing ip & ethernet adaptation for the metro ethernet market. In *Proc. European Conference in Networks and Optical Communications (NOC 2007)*, June 2007. Invited Paper.

[ea07b] S. Orlowski et al. SNDlib 1.0–Survivable Network Design Library. In *Proceedings of the Third International Network Optimization Conference (INOC 2007), Spa, Belgium*, April 2007. http://sndlib.zib.de.

[ea08] Fernando Solano et al. All-optical label stacking: Easing the tradeoffs between routing and architecture cost in all-optical packet switching. In *INFOCOM*, April 2008. Accepted for publication.

[FATW05] J. Farkas, C. Antal, G. Toth, and L. Westberg. Distributed resilient architecture for Ethernet networks. *Design of Reliable Communication Networks, 2005.(DRCN 2005). Proceedings. 5th International Workshop on*, page 8, 2005.

[FAW+06] J. Farkas, C. Antal, L. Westberg, A. Paradisi, TR Tronco, and V. Garcia de Oliveira. Fast Failure Handling in Ethernet Networks. *Communications, 2006. ICC'06. IEEE International Conference on*, 2, 2006.

[Fea] D. Fedyk et al. Gmpls control of ethernet pbb-te. Internet draft.

[IKM03] R. Inkret, A. Kuchar, and B. Mikac. Advanced infrastructure for photonic networks european research project. In Extended Final Report of COST 266 Action, ISBN 953-184-064-4, 2003. p. 21.

[INB+07] S.M. Ilyas, A. Nazir, F.S. Bokhari, Z.A. Uzmi, A. Farrel, and F.R. Dogar. A Simulation Study of GELS for Ethernet Over WAN. *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 2617–2622, 2007.

[KL00] Murali S. Kodialam and T. V. Lakshman. Minimum interference routing with applications to MPLS traffic engineering. In *INFOCOM (2)*, pages 884–893, 2000.

[MB76] Robert M. Metcalfe and David R. Boggs. Ethernet: distributed packet switching for local computer networks. *Commun. ACM*, 19(7):395–404, 1976.

[NNM+06] P.M.V. Nair, S.V.S. Nair, M.F. Marchetti, G. Chiruvolu, and M. Ali. Distributed Restoration Method for Metro Ethernet. *Proceedings of the International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies (ICNICONSMCL'06)-Volume 00*, 2006.

[PDV05] D. Papadimitriou, E. Dotaro, and M. Vigoureux. Ethernet layer 2 label switched paths (lsp). In *Proc. of Next Generation Internet Networks*, pages 620–621, April 2005.

[Pea] D. Papadimitriou et al. Generalized mpls (gmpls) rsvp-te signaling in support of layer-2 label switched paths (l2 lsp). Internet draft.

[Per00] Radia Perlman. *Interconnections Second Edition: Bridges, Routers and Switches*. Addison-Wesley, 2000.

[QMCL08] Jian Qiu, Gurusamy Mohan, Kee Chaing Chua, and Yong Liu. Local restoration with multiple spanning trees in metro ethernet. *Optical Network Design and Modeling, 2008. ONDM 2008. International Conference on*, pages 1–6, March 2008.

[Quo05] B. Quoitin. Topology generation based on network design heuristics. *Proceedings of the 2005 ACM conference on Emerging network experiment and technology*, pages 278–279, 2005.

[RKM+05] F. Ramos, E. Kehayas, JM Martinez, R. Clavero, J. Marti, L. Stampoulidis, D. Tsiokos, H. Avramopoulos, J. Zhang, PV Holm-Nielsen, et al. IST-LASAGNE: Towards All-Optical Label Swapping Employing Optical Logic Gates and Optical Flip-Flops. *Lightwave Technology, Journal of*, 23(10):2993–3011, 2005.

[RVC01] Eric Rosen, Arun Viswanathan, and Ross Callon. *Multiprotocol Label Switching Architecture*. IETF, January 2001. RFC 3031.

[San03] Ralph Santitoro. *Bandwidth Profiles for Ethernet Services.* Metro Ethernet Forum, Oct 2003. http://www.metroethernetforum.org/metro-ethernet-services.pdf.

[SCdO+07] Fernando Solano, Luis Caro, Jaudelice de Oliveira, Ramon Fabregat, and Jose Marzo. G^+: Enhanced traffic grooming in WDM mesh networks using Lighttours. *IEEE Journal on Selected Areas in Communications (JSAC)*, June 2007.

[SFDM05a] Fernando Solano, Ramon Fabregat, Yezid Donoso, and Jose Marzo. Asymmetric tunnels in P2MP LSPs as a label space reduction method. In *Proc. IEEE International Conference on Communications (ICC 2005)*, pages 43–47, May 2005.

[SFDM05b] Fernando Solano, Ramon Fabregat, Yezid Donoso, and Jose Marzo. A label space reduction method for P2MP LSPs using asymmetric tunnels. In *Proc. IEEE International Symposium on Computers and Communications (ISCC 2005)*, pages 746–751, June 2005.

[SFM05] Fernando Solano, Ramon Fabregat, and Jose Marzo. A fast algorithm based on the MPLS label stack for the label space reduction problem. In *Proc. IEEE IP Operations and Management (IPOM 2005)*, October 2005.

[SFM08] Fernando Solano, Ramon Fabregat, and Jose Marzo. On optimal computation of MPLS label binding for multipoint-to-point connections. *IEEE Transactions on Communications*, July 2008.

[SGNC04] S. Sharma, K. Gopalan, S. Nanda, and T. Chiueh. Viking: a multi-spanning-tree Ethernet architecture for metropolitan area and cluster networks. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 4, 2004.

[SMW02] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring isp topologies with rocketfuel. In *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 133–145, New York, NY, USA, 2002. ACM Press.

[SMY00] Hiroyuki Saito, Yasuhiro Miyao, and Makiko Yoshida. Traffic engineering using multiple multipoint-to-point LSPs. In *INFOCOM (2)*, pages 894–901, 2000.

[Wax88] B. M. Waxman. Routing of multipoint connections. *Selected Areas in Communications, IEEE Journal on*, 6(9):1617–1622, 1988.