Aalto University
School of Science and Technology
Faculty of Electronics, Communications and Automation
Department of Communications and Networking

Jukka Nousiainen

MANAGEMENT OF CARRIER GRADE INTRA-DOMAIN ETHERNET

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Espoo 31.3.2010

Thesis supervisor:

Prof. Raimo Kantola

Thesis instructor:

M.Sc. (Tech.) Olli-Pekka Lamminen

Author: Jukka Nousiainen

Title: Management of Carrier Grade Intra-Domain Ethernet

Date: 31.3.2010 Language: English Number of pages: 12+77

Department: Department of Communications and Networking

Professorship: Networking Technology Code: S-38

Supervisor: Prof. Raimo Kantola

Instructor: M.Sc. (Tech.) Olli-Pekka Lamminen

Internet is evolving from its role as a mere information provider to an ubiquitous infrastructure crucial to society. The current technologies running the majority of global Internet—IPv4 in addressing, MPLS as core transport and SDH as the physical transfer technology—have been long-lived. However, their dominance has started to diminish because a network technology common to all, Ethernet, has started to expand from local to metropolitan and wide area networks. Most enterprises and home users already use Ethernet in their LAN. Connecting these sites to MAN or WAN with the same technology is the logical next step in technology consolidation.

This has raised the demand for Carrier Ethernet services. However, internally they are still mostly provided with non-Ethernet technologies such as MPLS or SDH, because currently Ethernet lacks the necessary service assurance components. The real challenge in future internetworking is creating a Carrier Ethernet Transport (CET). With CET, any imaginable telecommunication service is delivered with a purely Ethernet based technology. When we have Ethernet in transport networks, it is no more a long stretch to a global, routed end-to-end Ethernet.

This thesis covers management of an intra-domain CET control plane. First, Carrier Ethernet services and technologies currently producing these services are analyzed. Second, requirements imposed to CET and current CET candidates are discussed. Third, network management standards and their alignment to carrier business is studied. After the background has been discussed, a control plane management system developed for the EU 7th framework ETNA project is introduced. The management system is capable of provisioning point-to-point and multipoint services and is controlled via a web-service -based northbound interface. The control plane is able to install the services as forwarding entries in a network processor -driven data plane developed at Ben Gurion University.

Keywords: Carrier Grade, Ethernet, Carrier Ethernet Transport, Intra-domain, Network Management

AALTO-YLIOPISTON TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN TIIVISTELMÄ

Tekijä: Jukka Nousiainen

Työn nimi: Operaattoritasoisen Ethernet-verkon hallinta

Päivämäärä: 31.3.2010 Kieli: Englanti Sivumäärä: 12+77

Laitos: Tietoliikenne- ja tietoverkkotekniikan laitos

Professuuri: Tietoverkkotekniikka Koodi: S-38

Valvoja: Prof. Raimo Kantola

Ohjaaja: DI Olli-Pekka Lamminen

Internet ei ole enää pelkkä tiedonlähde, vaan enenevässä määrin kriittisempi osa yhteiskunnan infrastruktuuria. Nykyiset Internet-palveluja tuottavat teknologiat—IPv4 osoitteistuksessa, MPLS siirtoalustana ja SDH fyysisenä välitysteknologiana—ovat alkaneet menettää valta-asemaansa samalla kun kaikille tuttu verkkoteknologia, Ethernet, on laajentunut lähiverkoista runkoverkkoihin. Maailmassa on miljoonia Ethernet-lähiverkkoja. Olisi kustannustehokaampaa toteuttaa myös näiden lähiverkkojen väliset siirtoyhteydet Ethernetillä.

Halu kustannustehokkuuteen ja teknologian konsolidointiin on tuonut esille tarpeen ns. operaattorikestoisille Ethernet-palveluille. Koska Ethernetistä puuttuu määrättyjä ominaisuuksia joita ilman on mahdotonta toteuttaa siirtoverkkopalveluja, näitä operaattori-Ethernet-palveluja on tuotettu toistaiseksi olemassaolevilla tekniikoilla, kuten MPLS:llä. Tulevaisuudessa todellinen haaste on luoda operaattoritasoinen, Ethernet-pohjainen siirtoverkkoteknologia, joka kykenee tuottamaan Ethernet-palvelujen lisäksi mitä tahansa muita tietoliikennepalveluja.

Tämä diplomityö käsittelee operaattoritasoisen Ethernetin hallintaa yhden runkoverkkoalueen sisällä. Työssä käydään läpi standardoidut operaattorikestoiset Ethernet-palvelut, teknologiat joilla palveluja tällä hetkellä tuotetaan, ehdokkaat tulevaisuuden Ethernet-siirtoverkkoteknologioiksi sekä keskeisimmät verkonhallintaan liittyvät standardit. Työn jälkimmäisessä puoliskossa esitellään Euroopan Unionin 7th Framework ETNA -projektia varten kehitetty verkonhallintajärjestelmä. Hallintajärjestelmä tarjoaa rajapinnan jonka kautta on mahdollista provisioida suojattuja Ethernet-palveluja kahden asiakasliityntäpisteen välillä, ja lisäksi lähetyspuita joissa kohteina on useampi asiakaspiste. Hallintajärjestelmältä tilatut palvelut viestitetään Ben Gurionin yliopiston toteuttaman, verkkoprosessoreilla toimivan välityskerroksen välitystauluihin.

Avainsanat: Operaattoritasoinen, Ethernet, Verkonhallintajärjestelmä, Verkon hallinta

Preface

Working in ETNA has been a very interesting time for the undersigned. When I started doing research about Ethernet transport in the beginning of summer 2008, I had just taken a network provisioning course which had big focus on Multiprotocol Label Switching. Back then it seemed like the world is an oyster for MPLS. However, throughout the following year and a half it became evident to me that the control plane of MPLS has deficiencies that make it too inflexible for delivering carrier grade transport services. In tandem, our team at ComNet was designing a transport oriented control plane which should be superior to an IP-based solution. The control plane architecture was designed by Marko Luoma, Olli-Pekka Lamminen and Raimo Kantola. Several people worked on the implementation. Oscar Santolalla modified the Quagga routing protocol suite with Carrier Ethernet extensions. Visa Holopainen crafted various approaches for calculating protected point-to-point and multipoint paths in the Path Computation Element of the control plane Management System. The Management System itself was designed and implemented by yours truly. Jouko Lehtonen coded the Web UI for the MS. Jan Gröndahl programmed a library which enables the control plane to communicate with the data plane using the ForCES framework. Taneli Taira designed management and transport service extensions to the RSVP-TE protocol, implemented a signalling daemon for this extended protocol and defined various frame types which the signalling can utilize for both point-to-point and multipoint service delivery. All the while Olli-Pekka steered the implementation in the big picture.

I wish to thank my supervisor, prof. Kantola, who has had the courage to see the deficiencies in current IP networks and strive to create something better. In addition I am much obliged to my instructor Olli-Pekka, who has relentlessly guided my work both during writing this thesis and while designing & implementing the control plane Management System. Futhermore, I wish to express kindest thanks to all the people in the Department of communications and networking I have had the pleasure of knowing and working with during the course of the past two years. Thanks should also go to all ETNA partners in BT, Ethos, NSN and Ben Gurion University, who were part of designing and implementing the ETNA architecture. I would also like to give thanks to my family for all their support.

Finally, I wish to express my utmost gratitude to Neetta for supporting me through thick and thin during the writing of this thesis.

In Helsinki, 31st of March 2010

Jukka Nousiainen

Contents

Abbreviations

802.1Q	IEEE Virtual LANs
802.1ad	IEEE Provider Bridges
802.1ah	IEEE Probider Backbone Bridges
802.1ag	IEEE Connectivity Fault Management
802.1Qay	IEEE Probider Backbone Bridges with Traffic Engineering
ATM	Asynchronous Transfer Mode
CAPEX	Capital Expenditures
CE	Control Element
CET	Carrier Ethernet Transport
CFM	*see 802.1ag*
E-NNI	External Network-to-Network Interface
ELS	Ethernet Label Switching *a.k.a VLAN XC*
ETNA	Ethernet Transport Network, Architectures of Networking
FCAPS	Fault, Configuration, Accounting, Performance, Security
FE	Forwarding Element
GFE	Generic Forwarding Element
IETF	Internet Engineering Task Force
I-NNI	Internal Network-to-Network Interface
ISP	Internet Service Provider
LAN	Local Area Network
L1	Physical Layer of the OSI Model
L2	Link Layer of the OSI Model
L3	Network Layer of the OSI Model
MAN	Metropolitan Area Network
ME	Maintenance Entity
MEF	Metro Ethernet Forum
MEG	Maintenance Entity Group
MEP	MEG End Point
MIP	MEG Intermediary Point
MPLS	Multiprotocol Label Switching
MPLS-TP	Multiprotocol Label Switching Transport Profile
MS	Management System
NE	Network Element
NMS	Network Management System
OAM	Operation, Administration and Maintenance
OPEX	Operational Expenditures
OSI	Open Systems Interconnection
OSS	Operational Support System
PCE	Path Computation Element
PBB	*see 802.1ah*
PBB-TE	*see 802.1Qay*
Q-in-Q	*see 802.1ad*
QoS	Quality of Service

RSVP-TE	Resource Reservation Protocol with Traffic Engineering
SDH	Synchronous Digital Hierarchy
SOAP	Simple Object Access Protocol (from v. 1.2 just "SOAP")
TE	Traffic Engineering
U-NI	User-to-Network Interface
WAN	Wide Area Network
VLAN	Virtual Local Area Network
VLAN XC, VXC	VLAN Cross Connect
VPN	Virtual Private Network

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Foreword

Ethernet has went through an immense transformation since its initial conception in the 1970s. Developed originally as a means for computers inside a limited geographical area to communicate with each other, Ethernet has spread out to be the most popular wired home and corporate network technology. In the 1990s Ethernet got standardized optical interfaces and Virtual Local Area Network (VLAN) capability. Most enterprises use normal or VLAN-tagged Ethernet in their LANs, and some even run storage area networks and various other applications on top of it.

This popularity of Ethernet in home and corporate networks has led to high volume production of Ethernet components, resulting in very low manufacturing costs. In addition, Ethernet standardization has always been able to increase transmission speeds when present-day interfaces have become too slow to accommodate all required traffic. This holds true even today[1] when Gigabit Ethernet has become a standard household item. Because consolidating network technologies has the tendency to cut down costs, enterprises are gradually requiring more and more site connectivity based on Ethernet. The demand for Ethernet services has initially been satisfied by providing them on top of non-Ethernet solutions such as Multiprotocol Label Switching (MPLS) Virtual Private LAN service. While such systems have very little to do with Ethernet, the services they offer are nevertheless sold with the moniker *Carrier Ethernet*.

At the same time Internet is evolving from its role as an information provider to an infrastructure that is crucial to society. Technologies running the majority of global internetworks are IPv4 in addressing, various platforms from Frame Relay to Asynchronous Transfer Mode (ATM) to MPLS in switching, and multiplexed or standalone Synchronous Digital Hierarchy (SDH) as physical transport. Thus we

[1] The IEEE P802.3ba Task Force, among others, is actively developing Ethernet to meet the 100Gbps mark. Their progress can be followed in `http://www.ieee802.org/3/ba/public/index.html`.

have a multitude of enterprises and Internet users being served by different technologies stacked on top of each other, as seen on the left-hand side of Figure 1.1. While this does provide what the end-users want, it is far from an optimal solution. Managing a multitechnology, multi-layer network infrastructure takes its toll on both operational and capital expenditures of an operator. In a worst case scenario each techonology requires a separate support system which has no capability of understanding data models or properties of other parts of transport service infrastructures. Each of these systems might have their own data storages and protocols for customer information, element management, problem ticketing and so on. A system built this way—each component so restricted to each other that functionality cannot be extended, and lacking interfaces for communicating with other systems—is called a stovepipe. When service infrastructure is built by cascading stovepipe systems it gets complicated to see simple things like which services are disrupted and which customers affected when a particular link gets broken. [3]

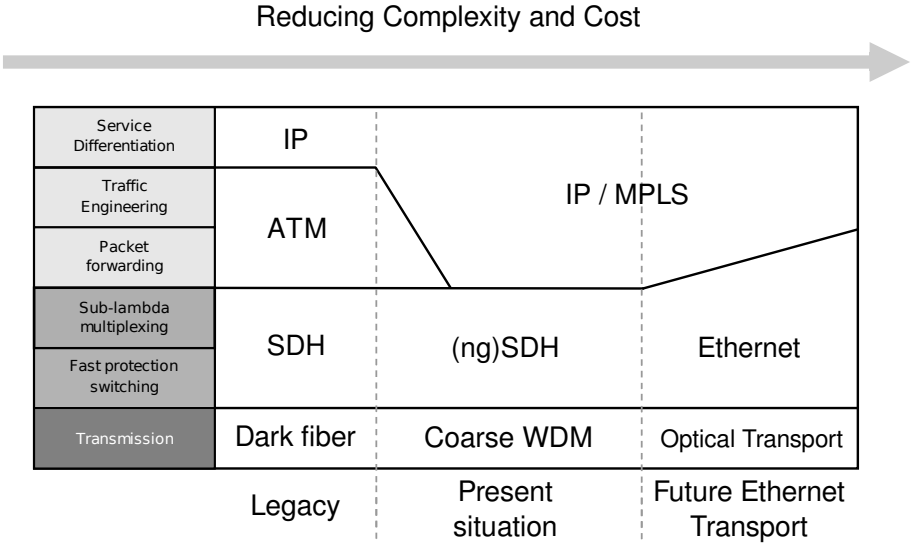


Figure 1.1: Packet switched transport network evolution. Adapted from [4].

Because Ethernet is already installed into millions and millions of sites, it would be natural to trash these multi-layered constructs and and start building an internetwork that would run on Ethernet completely from end to end. To illustrate this the right-hand side of Figure 1.1 shows the current packet-based transport incumbent, IP / MPLS, gradually starting to be phased off in favor of Ethernet. Unfortunately there are problems to be solved before this can happen. For one, Figure 1.1 does not by all means convey the full breadth of telecommunication services used in todays internetworks. To be able to completely replace a technology such as SDH, *all* existing services from landline calls to mobile base station backhaul to leased lines—basically any conceivable telecommunication service, should be running on top of this new Ethernet solution. In order to achieve this, Ethernet has to be transformed into a carrier grade service delivery platform—Carrier Ethernet

Transport (CET). This cannot be done without building new routing functionalities and doing modifications to the base Ethernet specification.

1.2 Thesis motivation and goals

It is hard to build Carrier Ethernet services with control plane functions being unilaterally local to each network node as in conventional Ethernet. This stems from the fact that traditional Ethernet is not designed to converge in a controlled, predictable manner because it uses flooding to populate forwarding tables and spanning tree mechanisms to form a loop-free network topology. Relying on optimistically finding best paths for traffic via dynamic negotiation, or transmitting to all ports if a destination is unknown, is unacceptable for carrier grade transport networks—no unwarranted frames should enter the network, especially if services are not of constant bit rate and thus congestion can occur. A carrier grade Ethernet transport network needs a centralized management system to be able to provision services over optimal paths and to keep forwarding stable even when disruptions happen. This thesis will concentrate on such a management system, as will be stated in the following problem definition. [3]

The purpose of this thesis is to study management of a single transport domain that provides Carrier Ethernet services and to build prototype software that implements at least a subset of this management functionality. The system in question is developed for the EU 7th framework Ethernet Transport Networks, Architectures of Networking (ETNA) project to manage control plane operations.

The focus is in CET control plane functionality. Items of research include how we control and monitor the topolology and utilization of the network, and also how services can be provisioned with respect to current utilization level. What is also covered is how specific service characteristics are taken into account when computing and signalling paths for services.

While CET internetworking is an important subject, focus of this thesis is purely on intra-domain CET, which means that all network elements within the control plane are considered the property of a single operator. In addition, all traffic and network interfaces (core- and customer-facing) are considered trustworthy. With respect to scalability, we ensure that control plane APIs warrant dynamic addition of nodes into the domain. While bias is mostly in technical control plane management, an additional goal is to study how network management functions performing tasks above can be structured into the carrier organization.

What is ruled out

Creating transport service billing and charging processes on top of this control plane solution is left out of scope. Comparing various company strategies, business models and their alignment with particular network management paradigms is a

very broad subject and is not covered. As the scope is intra-domain, provisioning of inter-domain CET services is ruled out. ETNA control plane northbound interface allows building inter-domain services when coordinated by an external provisioning system, but this scenario is not discussed in this thesis.

Management and data plane operations are discussed only when their importance to the control plane operation requires it. Thus, matters related to data plane's physical Ethernet interfaces and internal architecture, detailed description of the forwarding mechanisms and so on are primarily truncated. They will be covered only if it is hard to explain certain aspects of the control plane without them, eg. framing will be introduced if it contains information that needs to be managed from the control plane. Scalability issues that would require simulation—for example rolling an ETNA system globally—are not covered. Since all traffic is regarded as non-malevolent, security issues are not considered.

Most of todays network management solutions are proprietary software developed by carriers themselves. Some of them also use proprietary methods of communication with other systems and expressing data. The architecture of such systems is usually a business secret, and it is not realistic to try to gather information about them for this thesis. Operator-manufactured management systems will therefore not be discussed. [30]

To allow modularity, the Management System should be as agnostic as possible regarding the protocols used for communication. Therefore topology discovery & signalling protocols used between control plane elements, and on the other hand control plane $\Longleftrightarrow$ data plane communication protocols are not discussed in a detailed level. These subjects are covered thoroughly in [14, 41, 42, 38, 6, 7, 25].

1.3 Structure

In the final section of this introduction some network management terminology and concepts used throughout this thesis will be defined. In Chapter 2 we will start by first discussing the common Ethernet technologies currently used in LANs, continue to the services that have been defined for connecting these LANs, explain the history of transport networks currently delivering these services, and move on to describing several of Carrier Ethernet Transport candidates. Transport network history is important because CET will never be rolled out overnight - it must exist alongside current transport network deployments for quite some time. In Chapter 3 we will briefly introduce the transport service provider as an organizational unit, after which we will study existing literature and standards regarding network management systems and interfaces. This will give us a better understanding how to integrate the control plane into a provider organization. Chapter 4 presents the architecture and functionality of the control plane management system that was implemented in ETNA. In Chapter 5, this system is evaluated with respect to the background information covered in Chapters 2-4. Chapter 6 lines out some conclusions and gives directions for future research. The thesis is concluded with a general

summary.

1.4 Key Terminology

In the following we will define terminology related to networking and network management which are essential to understand in order to comprehend the framework of this thesis.

Faults

A fault can be considered as any disruption in service the carrier is providing with its network. Key terms related to faults are:

Availability
The portion of time the system is functioning.

Mean time between failures (MTBF)
The average time between faults experienced by the system.

Mean down time (MDT)
The average time that a system is non-operational.

Mean time to repair (MTTR)
The average time required to repair a failed system.

Naturally a carrier wants to maximize availability & MTBF and minimize MDT & MTTR for any given service that is sold.

Internet Service Provider

An Internet Service Provider (ISP) sells IP connectivity and other IP-based services to personal and corporate clients or other ISPs. An ISP itself may also buy IP connectivity from another ISP with a transit agreement. Both of these relations can also be formed with a so called peering agreement, where both parties exchange the routes and traffic of their own customers but neither charges the other as long as the traffic is symmetrical enough.

This thesis covers mostly tranport networks instead of IP networks where peering models are not applicable and most contracts are of transit-like nature. However, if a transport service provider doubles as an ISP and runs IP traffic on top of its backbone, it must additionally take characteristics of IP services into account when provisioning the network.

Local Area Network

A computer network contained within a small geographical area. An example would be any home network, or for example a small-to-medium sized company network or a school network.

Metropolitan Area Network

A computer network which extends over a moderately large geographical area such as a city, technology park or a well-spread university campus. The purpose of a MAN is to interconnect multiple LANs.

Northbound Interface

A Northbound Interface of a particular computer system is an interface which provides controlling capabilities and information about the system to upper, higher order systems.

Operation, Administration and Maintenance

Operation, Administration and Maintenance (OAM) is a general term used for a tool for maintaining services in telecommunication networks. There are various OAM technologies and standards, but the common factor is that all are used to perform measurements or troubleshooting problems in carrier networks.

Operational Support Systems

An OSS is a program or a set of software that is used to configure and maintain networking equipment. This might include, for example, traffic engineering paths in a congested domain, processing counters for charging & billing, or populating edge switches' forwarding tables with customer site information. In the most complex cases a single OSS will configure network equipment to process information on physical, link, network, transport and even application layer of the OSI model. The purposes for running an OSS can range anywhere from pure technical management to service assurance, customer authentication or wiretapping.

Quality of Service

Quality of Service (QoS) is a term for the quantitative and qualitative characteristics necessary to achieve some level of functionality and end-user satisfaction with a given service. In transport network context it is usually defined with attributes strongly related to transport of raw data such as packet loss, delay, jitter,

set-up and tear-down times, and so on. QoS is most often used for measuring end-to-end health of a particular service but it is not uncommon to perform QoS measurements in single parts of network such as a junction between two providers. [43]

Routing

Routing is a process of gathering, pre-processing and making use of information for the purpose of directing traffic in network nodes from sources to destinations, and doing this in such a manner that the constraints for the traffic demands are met. Routing is a fundamental capability needed in all kinds of networks but appears in different forms in different technologies. Routing is needed everywhere where some kind of addressing scheme is in place and end hosts contained within a specific network need to send information to end hosts possibly external to that network.

Service Level Agreements

A Service Level Agreement (SLA) is a written, legally binding contract, usually made between two companies. In telecommunication context the companies are most commonly a service provider and a customer. The purpose of an SLA is to define quantifiable and measurable parameters or constraints which monitor that the service in question is provided in agreed manner. SLAs are important in situations where legal regulation does not govern the service in question or does so ambiguously. The parameters in an SLA can be for example:

Costs for service over a given time period

Connection types such as link type and speed, synchronization

Capacity for example constant bits per second or over a measurement interval

QoS characteristics such as maximum jitter or round-trip time

Corruption e.g. maximum bit error rate

Fault measures such as availability or MTTR

Security level against compromisation by a third party

Responsiveness such as MTTR, set-up times, tear-down times, synchronization delays

Sanctions to be given if the SLA has been broken.

QoS-based SLAs are almost mandatory for transport services because transport networks usually carry mission-critical data. An important area of carrier SLAs

is defining who does the actual measurements and in what parts of the network and how they are performed. In IP networks most traffic is best effort and recovery is either based on slow convergence of the internal routing protocol of the ISP experiencing the fault or searching for another prefix to circumvent that ISP's autonomous system. IP-based SLAs do not have strong connection to any QoS attributes, but instead focus on how fast the fault should be repaired and how the customer is informed about repairs. [43]

Service provider

A service provider is a company buying telecommunication capability from one or multiple transport or internet service providers and selling the combined result as a single service. For example a service provider might have customer that needs to connect two sites in separate countries, while no transport service provider operates in both of them. The service provider would buy transport bit pipes from domestic transport service providers up to their common Network-to-Network Interface (NNI) and sell the aggregated service to the customer.

Southbound Interface

Southbound Interface is an interface which is used to access lower layer elements from a particular computer system. Most often this is used to query information from the underlying system or give control messages to it.

Traffic Engineering

Traffic Engineering (TE) in telecommunication context is a field where statistical tools, queuing theory, simulations and measurements are used to optimize communication networks to function with minimum cost. In voice networks the concept has been succesfully understood and applied for decades since voice traffic is very predictable both in growth and usage. In packet switched networks the application of TE gets harder as there is no inherent notion of a call or a video stream. However, the purpose is still the same: to provide reliable services with while keeping costs low.

In transport networks most traffic has a pre-provisioned destination, so in theory they are easier to traffic engineer. However, transport networks often have guarantees such as a constant bit rate must be maintained for a given service which complicates the issue. IP networks, on the other hand, do not usually employ QoS schemes or give guarantees for traffic delivery, so for that part IP TE is easier. Their downside is that it is a very resource consuming and error-prone process to identify a particular "flow" which should get differentiated forwarding priority.

Transport Service Provider

A transport service provider is a company that sells physical or link layer connectivity. One could for example purchase a transport service to connect private branch exchanges or Ethernet switches on separate customer sites. For the purposes of this thesis words carrier, operator, transport provider and transport service provider are used synonymously. In the same fashion carrier networks, operator networks and transport networks are used synonymously.

Virtual Private Network

A Virtual Private Network (VPN) is a slice of a physical network which connects a given set of sites. A VPN usually exists on L2 or L3 but leased line -type solutions can also be considered as L1 VPNs. From the client site point of view the VPN is either a virtual Ethernet switch which possibly recognizes the clients VLANs, or a router with site subnetworks only one hop apart.

Wide Area Network

A Wide Area Network (WAN) is a computer network that spans a very broad geographical area, even over national or continental borders. WANs are created for example to interconnect corporate MANs or LANs together for the purposes of setting up a consolidated intranet of all company sites despite of their location, or on the other hand to transport general IP traffic. It is important to note that WAN does not always equal the internet even though many router vendors use this term for the area outside LANs. The (possible) addressing used in customer ends of a WAN has only semantics local to that specific WAN. End stations behind a WAN may as well communicate only with MAC addresses, which currently do not have global meaning. In contrast, IPv4 or IPv6 addresses have fully global semantics as long as they are not private addresses[2].

[2]IPv4 and IPv6 address spaces contain private, globally unroutable subnets defined in Request For Comments (RFC) 3330 and RFC5156

Chapter 2

From Ethernet to a Transport Service

The purpose of this chapter is to give an overview about currently installed Ethernet network technologies and on the other hand explain how their adoption in high volumes has lead to the demand of Carrier Ethernet services. Thereafter the discussion proceeds to Carrier Ethernet Transport and the expectations imposed on several CET technologies.

2.1 Current LAN Deployments

We will start by reviewing the fundamental concepts behind two of the most widely used LAN technologies, "standard" Ethernet and IEEE 802.1Q.

Standard Ethernet

Ethernet was invented between 1973-1975 by a group of researchers working for the Xerox Palo Alto Research Center. The initial purpose was to offer connectivity between computers residing in different parts of the research center, enabling them to transfer information. The traffic which Ethernet was designed to carry was occasional, non-uniform bursts. [29]

The physical medium used in original Ethernet was copper wire, with which all the hosts were connected to each other. This also meant that if at any given time a host is transmitting into the so-called *Ether* formed by the wires, another host could easily interrupt this transmission by also starting to transmit. This is why the shared physical medium in vanilla Ethernet is frequently called a collision domain. At the time, this problem was solved by developing a mechanism for detecting collisions and recovering from them—Carrier Sense Multiple Access with Collision Detection (CSMA/CD). The initial 3 Mbps proof-of-concept network was seen to be a success and an alliance consisting of Digital, Intel and Xerox (DIX) was formed to

draft Ethernet protocol version 1.0, which offered transmission speeds of 10 Mbps. A revised version, 2.0, was later used as a basis for the first IEEE Ethernet standard, 802.3. DIX and 802.3 are nowadays the most commonly used LAN framing formats in the world. [29, 32]

As can be seen from Table 2.1, an Ethernet frame itself has quite simple structure. The destination and source MAC addressess distinguish the recipient and the sender, respectively. Only exceptions to this are some particular broadcast, multicast and vendor-specific destination addresses[1]. Two octets following the source address contain the EtherType (a field stating what kind of payload is to be expected) in the case of a DIX frame or a Length field in case of a 802.3 frame. To avoid clashes, only Ethertypes above 600 are assigned by IANA since Ethernet 802.3 frames are 1500 = 0x05DC octets long at maximum. The Cyclic Redundancy Check (CRC) or alternatively Frame Check Sequence (FCS) field is used to check the integrity of the frame.

Table 2.1: Structure of an Ethernet frame.

	octets
Destination MAC address	6
Source MAC address	6
EtherType or Length	2
Payload ⋮	46—1500
CRC / FCS	2

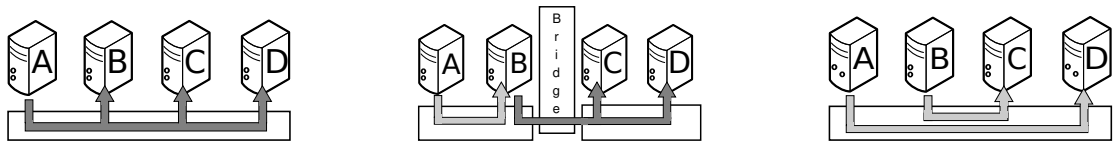
The most important concepts related to traditional Ethernet are hubs, bridges and switches. End hosts are connected to a hub in star topology, in which all frames are broadcast to all connected hosts as can be seen from Figure 2.1(a). This easily results in congestion even with quite moderate traffic if the transmit interval is small enough. Bridges and switches try to remedy this.

A bridge's main purpose is to offer traffic control by inhibiting frames from reaching network sections where they don't belong. A bridge sits between collision domains, and learns the source MAC addresses of each domain. When a bridge receives a frame, its destination address is compared with learned MAC addresses. If the recipient is in the same domain as sender the bridge does nothing. If the recipient is in another collision domain the bridge forwards the frame to that domain. Only if the destination MAC address is not known at all the frame is broadcast forward

[1] A list of reserved addresses can be found at `http://www.iana.org/assignments/ethernet-numbers`.

to adjacent domains, as can be seen from Figure 2.1(b). This increases the available network capacity per each domain as broadcasts are more limited and collisions occur less frequently.

A switch is a special case of a bridge, because each of its ports contain an isolated collision domain and is usually connected to only one device. Therefore there is no need for CSMA/CD at all as long as only one host is connected per port; all "collision" handling is done in the switch backplane with queuing. The switch always forwards frames without disruption to the correct destination if it is known. Because Ethernet devices flood frames with unknown recipients and can deploy spanning tree protocols to detect loops, Ethernet networks are easy to both build and expand. Ability to construct LANs in a plug-and-play manner is probably the most cogent reason why Ethernet has experienced such a huge success.



(a) Hub operation. Frames are broadcast to all ports.

(b) Bridge operation. Frames destined to same domain are not broadcast to other domains.

(c) Switch operation. Frame is sent to exact port where the recipient is connected.

Figure 2.1: Transmission methods of traditional Ethernet devices.

Tagged Ethernet

The size of local area networks grew steadily alongside the growth of Internet in the 1990s. Eventually scalability issues and lack of hierarchy in conventional Ethernet started to become an issue for larger organizations. This was remedied by dividing the network into virtual segments.

IEEE standard 802.1Q defines a 32-bit field that is embedded into the Ethernet frame in order to produce Virtual LANs (VLANs). The purpose of VLANs is to allow constraining a physical LAN topology into logical network segments. For example, a company may have a dedicated VLAN for Research and Development and another one for the Sales department, both of which are transported over the same physical Ethernet network. 802.1Q-enabled switches forward traffic based on the VLAN tag and destination MAC address. Because broadcasts are never sent from one VLAN to another, departments in separate VLANs are unable to communicate unless they are joined with a router or a switch configured to forward traffic between the departments' VLANs. In addition to introducing an overlay network on top of Ethernet, this also creates a layer of security. Bandwidth utilization in a network incorporating VLANs is also better due to limited scope of flooded broadcast and multicast traffic. [17, 37]

VLAN information, which can be seen from the greyer cells in Table 2.2,

Table 2.2: Structure of an IEEE 802.1Q frame. Darkest section indicates the VLAN tag, with which VLANs are managed.

				octets
Destination MAC address				6
Source MAC address				6
16-bit TPID	3-bit PCP	1-bit CFI	12-bit VLAN ID	4
Payload EtherType				2
Payload ⋮				46—1500
CRC/FCS				2

is embedded between the source MAC address and the payload EtherType. The first field is the Tag Protocol Identifier (TPID) Ethertype. The following fields describe the Priority Code Point (PCP) and Canonical Format Indication (CFI), which can be used for traffic shaping and Token Ring compatibility, respectively. From management perspective the most iteresting field is the 12-bit VLAN Identifier (VID) or "tag". VLAN network management essentially means configuring for each switch and for each port what tags are accepted upon reception, what sort of tagging (if any) is applied to untagged frames and what other VLANs the port in question belongs to. This sounds simple at first but steadily becomes harder the bigger the network and the more heterogenous the equipment base is, especially if one has to physically access the service console on each switch to configure the VLANs.

802.1Q is just as plug-and-play as normal Ethernet with respect to MAC learning and spanning trees. They are both technologies that scale well in small-to-medium size enterprise or even large enterprise networks. However, they are not suitable for carrier grade backbone or access networks where flooding and customer MAC learning are unwanted features. A large operator can easily serve millions of end stations, and learning all of their addresses would require an amount of memory most switches do not usually have. Additionally, 802.1Q has only 12 bits for VLAN identifiers, which allows for $2^{12} - 2 = 4094$ VLANs to exist inside a single domain (VLANs 0 and 4095 are reserved). Transport service providers can have thousands of corporate customers, each having multiple VLANs. [12, 37]

2.2 Carrier Ethernet

The ability to adopt Ethernet in high volumes with low costs and the fact that Ethernet LANs are easy to both install and extend have created demand for Ethernet MAN and WAN transport. Because transport service providers are constantly searching for the lowest euro per bit, Ethernet is a very viable candidate for a low-cost transport service delivery platform. This, however, will not happen until Ethernet service assurance tools have reached maturity and lower layers of Ethernet have been modified to allow synchronized services and carrier grade resiliency. In the meantime service providers will be offering Carrier Ethernet with whichever technologies they can use to meet demand with reasonable CAPEX and OPEX. In the absence of a dominant Carrier Ethernet Transport technology the standardization of Carrier Ethernet *services* has nevertheless begun.

The Metro Ethernet Forum (MEF) has defined Carrier Ethernet to be *"A ubiquitous, standardized, carrier-class network and service defined by five attributes that distinguish it from familiar LAN-based Ethernet"*:

- Standardized services
- Scalability
- Reliability
- Service management
- Quality of Service.

By this definition Carrier Ethernet is a set of certified network elements which offer Ethernet connectivity services over physical Ethernet or legacy transport technologies. In fact, any service—Ethernet or not—claiming to be carrier grade should contain these properties in some form. For example Ethernet-over-MPLS-over-SDH or Ethernet-over-SDH setups usually fullfill carrier grade requirements because the underlying physical transport, SDH, can be built in extremely resilient topologies offering sub-50ms failover times. [24, 39]

Next we will discuss the interfaces and standards defined for Carrier Ethernet. After this we will introduce some important Ethernet service assurance aspects. Lastly, a provider bridging solution for encapsulating VLANs is introduced because its framing is useful in conveying Carrier Ethernet services.

2.2.1 Interfaces

MEF has defined two standard interfaces for Carrier Ethernet:

User-to-Network Interface (U-NI) is a demarcation point between the customer and the service provider. It is the exact port on the provider equipment where the customer responsibility ends and the provider responsibility begins.

Network-to-Network Interface (NNI) is a term describing interfaces between Carrier Ethernet operator equipment. An *Internal NNI* (I-NNI) defines interfaces internal to a carrier domain. An *External NNI* (E-NNI) is a demarcation point between two different Carrier Ethernet networks, where EVCs of the first operator are translated to EVCs of the other.

Various service, OAM, protection and QoS parameters have been defined for configuring the Network Interfaces. Example service parameters include link type, MTU, rate and CoS preservation. QoS parameters include delay, loss and availability in addition to bandwidth profiles defined for various code points.

2.2.2 Services

Using the interfaces mentioned above, MEF has defined any Ethernet service between two or more UNIs as an *Ethernet Virtual Connection* (EVC). Currently three EVCs are defined:

E-line or point-to-point connection between two UNIs. E-lines currently dominate the market. This is due to the fact that the most needed Ethernet service is interconnecting two corporate sites running Ethernet.

E-tree or point-to-multipoint connection between two or more UNIs. An E-tree is usually defined as a hub-and-spoke arrangement where the spokes can speak to the hub but not to each other. There's currently some issues related to E-trees; if one uses them only unidirectionally for, say, a TV stream service from source to leafs, the carrier would probably like to reserve no reverse capacity from leaves to source. How can the customer then send OAM probes upstream to inform the sending station about a possible service disruption?

E-LAN or multipoint-to-multipoint connection between three or more sites. E-LAN management raises interesting Traffic Engineering questions not present in E-lines or E-LANs. One is how the stacked E-trees that form an E-LAN should share links. If a client requests a 1 megabit/second constant bit rate E-LAN for four sites which yields an overlapping 2-megabit full duplex reservation in some particularly central link, should he pay four times for the service? The management system has to be able to address such questions in computing paths for such a request. Most such billing policies are defined by carrier strategy and the network management system must be able to adapt to different policies.

These are basically the services any operator selling Carrier Ethernet today is providing. Figure 2.2 contains an example of each service. EVCs are classified into two categories. Port-based EVCs utilize a single service instance per UNI, which means that the provider network gives a dedicated resource for this service. VLAN-based EVCs on the other hand contain service multiplexes where several service

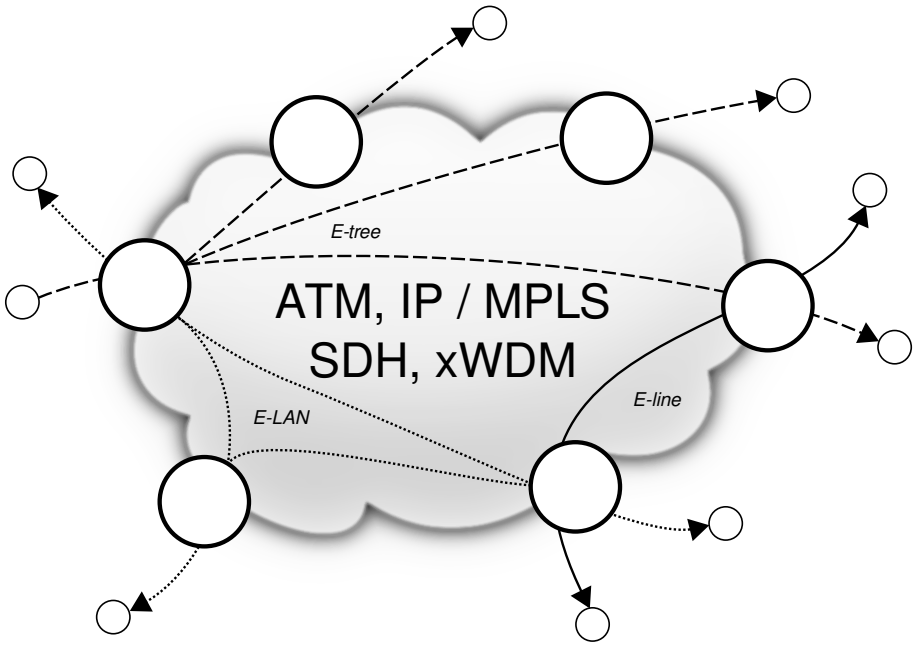


Figure 2.2: Current Carrier Ethernet services as defined by the MEF. The services do not limit the underlying transport infrastructure, so they are provided with various technologies.

instances are contained in the UNI, leading to a shared network resource. A port-based E-line is called an Ethernet Private Line (EPL) whereas a VLAN-based E-line is called an Ethernet Virtual Private Line (EVPL), and the same nomenclature is used for E-trees and E-LANs. [32, 39]

2.2.3 Service Assurance

Ethernet needs service assurance in order to be a carrier grade transport platform. MEF has emphasised that Carrier Ethernet services must be stringently monitored and measured in order to react to faults. The customer must be able to verify whether or not the purchased service is actually delivered as agreed, which can only be achieved by network measurements. There are two forms of measurements: active and passive. Passive network measurements require some sort of synchronization source, which raises additional obstacles. Active measurements are more common, but as they generate additional traffic in the form of test probes the measurement itself can interfere the service. This is accounted to some point in the test probe processing algorithms but is still good to keep in mind if measurements don't seem to correlate to actual utilization. [43, 15]

There are currently several finished standards for service assurance and miscellaneous Ethernet OAM functionalities:

IEEE 802.3ah or *Ethernet First Mile (EFM)* is an active measurement tool

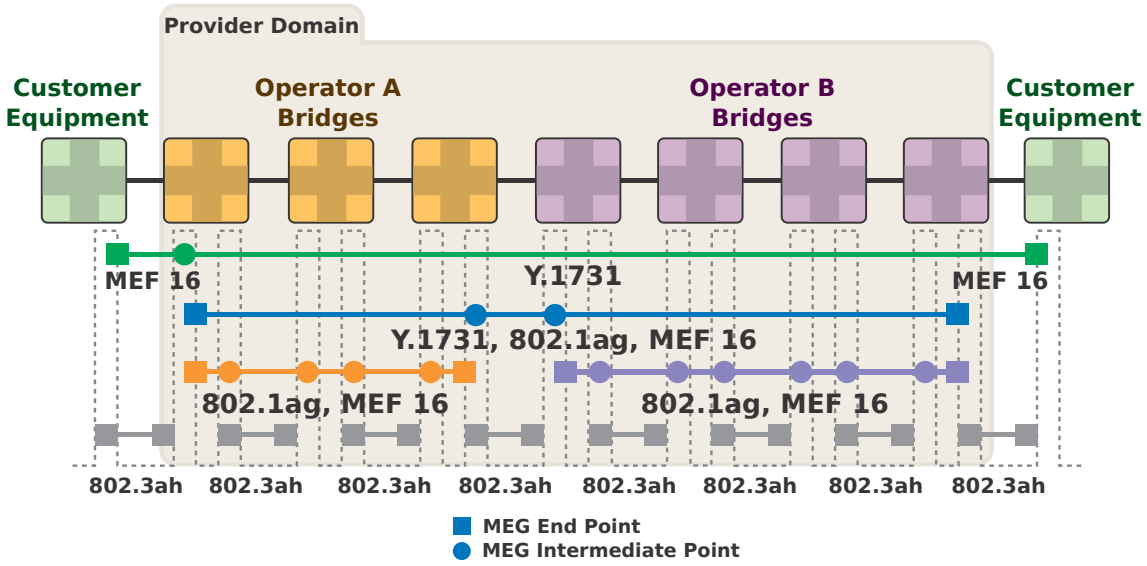


Figure 2.3: All currently ratified Ethernet OAM standards monitoring a Carrier Ethernet service. Adapted from [19].

used to monitor the connectivity of one physical link between two network nodes. Ethernet OAM PDUs defined in it provide functionality such as device discovery, remote failure indication, remote loopback and monitoring.

IEEE 802.1ag or *Connectivity Fault Management (CFM)* is set of tools for detecting, verifying and isolating faults on a per-VLAN level in bridged provider networks. This is achieved by various continuity check, traceroute, loopback and Alarm Indication Signal (AIS) messages.

ITU-T Y.1731 *"Requirements for OAM in Ethernet Networks"* is a UNI-to-UNI OAM standard used to maintain end-to-end services. Largely identical to 802.1ag, it defines OAM specific frame formats to use for measurements over service paths spanning over any kind of Ethernet topology but especially inter-domain. The biggest difference to 802.1ag is that Y.1731 supports performance monitoring and has more transport oriented focus.

MEF 16 or *Ethernet Local Management Interface (E-LMI)* enables the management plane of customer equipment to query the management plane of the provider edge equipment. Customer equipment can obtain information about status and attributes of Carrier Ethernet services. It operates over the UNI, and allows to some extent automatic configuration of services between customer and provider equipment because it can notify the customer about EVC creation, deletion and overall status. [44]

In an ideal situation all these tools work together to guarantee total end-to-end service assurance, as depicted in Figure 2.3. While E-LMI and 802.3ah are the

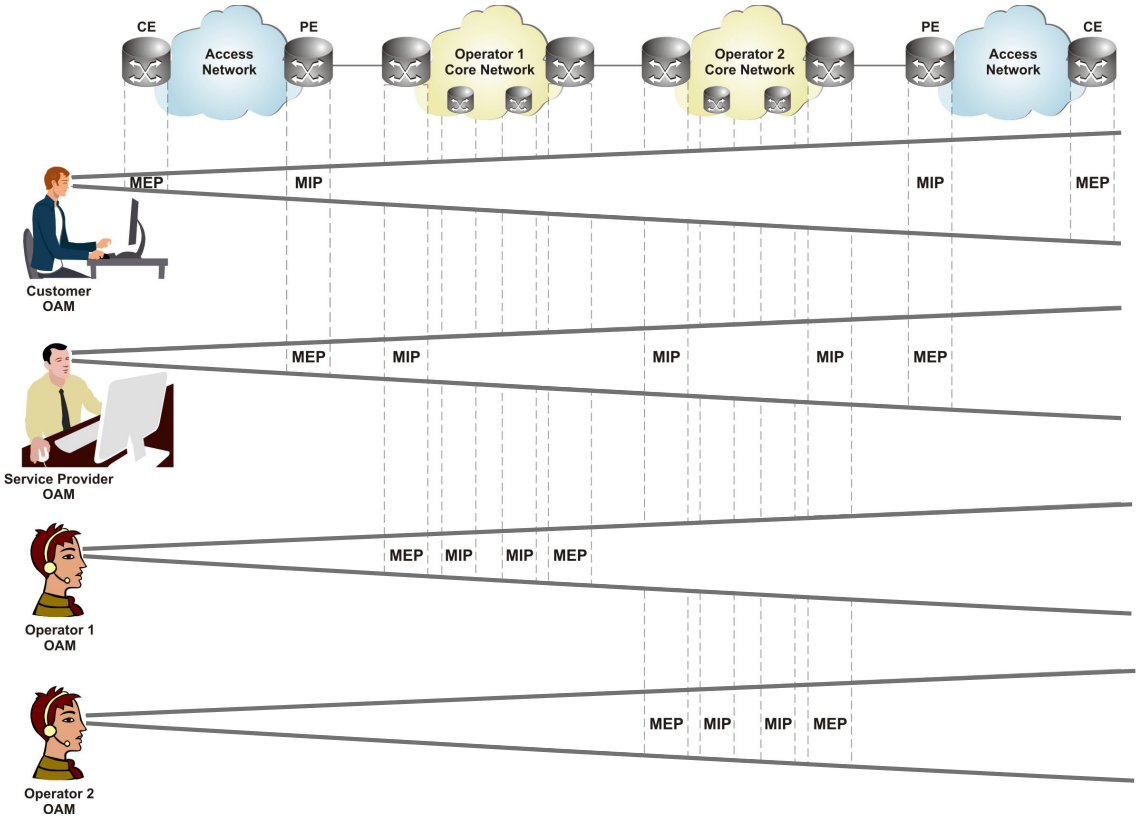


Figure 2.4: Multi-perspective management of maintenance endpoints. Services need to be inspected on many levels, thus OAM also needs to work on many levels. Notice that Operator 1's network does not support service provider MIP on other edge device. The service provider gets OAM information only between the MEPs and MIPs it has access to. [19]

most interesting OAM tools for intra-domain services, Y.1731 and 802.1ag can be used in intra-domain provisioning as well. Important concepts in Y.1731 are the so called Maintenance Entity (ME), Maintenance Entity Group (MEG), MEG End Point (MEP) and MEG Intermediary Point (MIP). A ME is basically any kind of network between two UNIs. A ME Group is an association of several MEs. Most importantly, a MEP is an entity which generates test probes into the MEG. A MIP is an entity which relays, monitors and possibly reacts to test probes flowing through it, but never generates any probes. Measurements between MEPs are generated on eight different levels, from 0 to 7. Levels 0-2 are reserved to carriers, 3 and 4 to service providers and the rest to customers. An example of the layering of MEPs and MIPs can be seen in Figure 2.4, where each management person has a different view on exactly the same service.

In light of this thesis the most relevant thing is managing maintenance entities or groups on levels 0-2. As Figure 2.5 shows, these levels mostly interact directly over UNI, ENNI and with other MEPs such as an MPLS MEP.

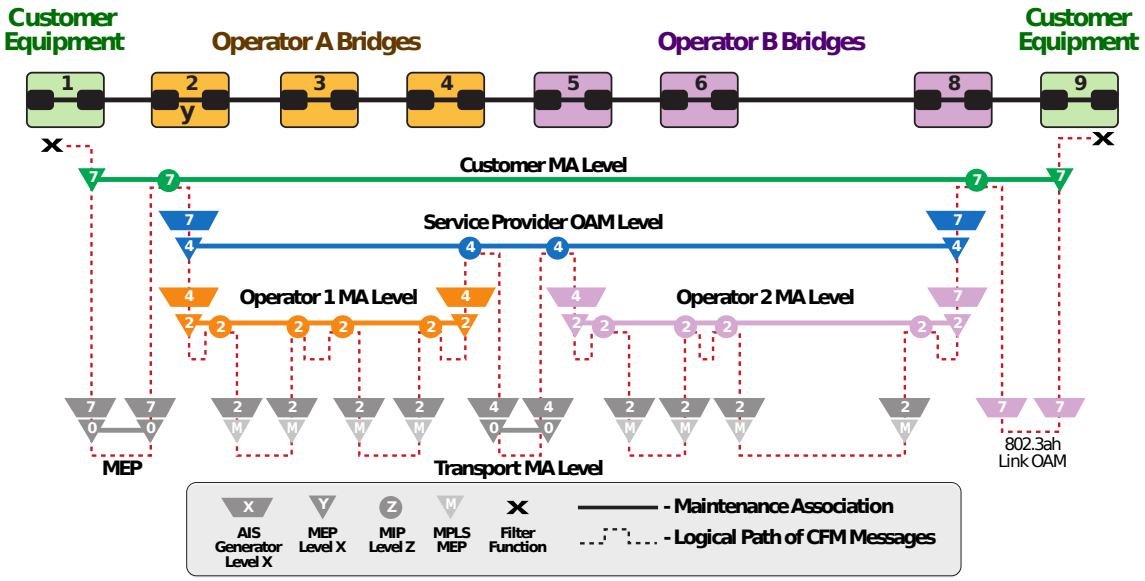


Figure 2.5: Connectivity Fault Management traversal through a two domain service provider network. [19]

Ethernet Access with Provider Bridging

An important concept related to standardized Ethernet services is IEEE 802.1ad or *Provider Bridges* (PB). PB was created to overcome limitations imposed by the 802.1Q address space. Depicted in table 2.3, 802.1ad defines another VLAN tag that is stacked before the 802.1Q tag.

Table 2.3: Structure of an IEEE 802.1ad (Q-in-Q) frame.

	octets
Destination MAC address	6
Source MAC address	6
Service tag (S-VLAN)	4
Customer tag (C-VLAN)	4
EtherType	2
Payload	46—1500
⋮	
CRC/FCS	2

Because of the stacked VLAN tags PB is often called Q-in-Q. The outer (S-VLAN) tag is used by the service provider and the inner (C-VLAN) tag space can then be used by customers however they please. S-VLAN solves the problem of administering multiple VLAN spaces by separating the customer VLAN space from the providers service VLAN space. A service sold to a specific customer can now be referenced solely with the S-VLAN tag. [10, 11, 40]

However, 802.1ad is no better for carrier grade operation than 802.1Q. 802.1ad still allows for only 4094 service instances. Additionaly, MAC addressing is still visible within the S-VLAN space, which means that the provider's bridges must learn heaps of customer addresses and forward based on them. While currently it is unclear if Provider Bridging will get widespread adoption, Q-in-Q *framing* itself has been taken into use in various networks providig Carrier Ethernet services.

A common practice for representing a MEF-defined EVC inside a provider core is to convert the single tagged 802.1Q frame into a double tagged 802.1ad frame on the UNI and use the outer tag as the identifier for a particular EVC inside the core, irregardless of Provider Bridges being present or not. Some off the shelf 802.1Q switches can forward frames even if they exceed the "normal" MTU by some octets as long as the outer tag is denoted with the standard 802.1Q Ethertype. In this case very cost-effective access networks can be built if (de)marcation is applied on edges with some kind of low-cost tagging devices and normal 802.1Q switches forward based on the outer tag inside the core. [10]

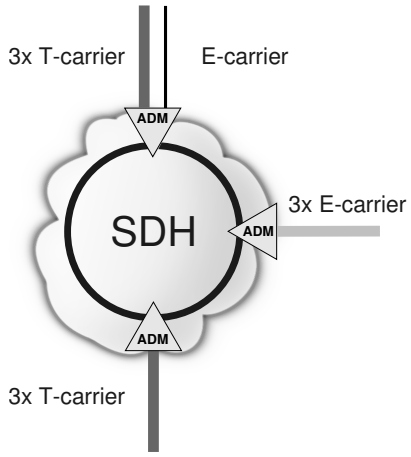
2.3 Current Carrier Ethernet Deployments

In the following we will discuss the majority of technologies running current Carrier Ethernet networks, and more importantly what makes them unsuitable for providing CET services.

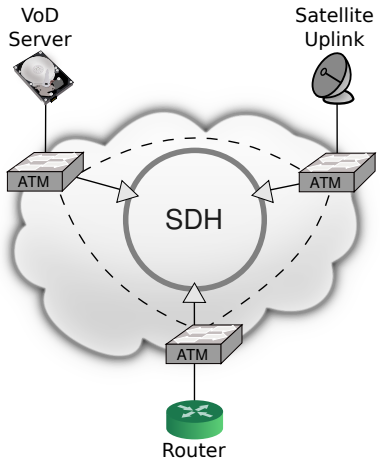
Synchronous Digital Hierarchy

In an SDH network all the channels are kept in sync with a single reference clock signal, and the framing repeats at a consistent 125 microsecond interval. When deployed in a suitable ring topology, SDH is able to offers resilience and fast (sub-50ms) fail-overs. Because SDH is synchronized by nature, offering services with strict latency bounds and resiliency requirements is easy and predictable. These properties have become the industry standard in delivering transport services, and nothing less should be expected from CET. [32]

An SDH network can be built into a point-to-point, mesh or ring topology. A carrier network with stringent requirements is usually ring-based because it provides best fail-over times. An example ring topology can be seen in Figure 2.6(a). The frames exchanged between the Add-Drop Multiplexers (ADM) are general containers for any kind of traffic, be it voice or data. They do not contain addresses, so



(a) An example SDH infrastructure transporting TDM traffic.



(b) ATM switches on top of an SDH core, switching traffic between a router, a Video on Demand - server and a satellite uplink. Logical ATM topology is denoted by dashed lines.

Figure 2.6: SDH-based transport networks.

switching is instead based on physical location of the interface and the location of a timeslot in a frame. SDH is usually managed with a protocol called Transaction Language 1 (TL1), which needs middleware such as a Common Object Request Broker Architecture (CORBA)-generated program to perform interpreting of external instructions to TL1 ones. TL1 syntax is human- and machine-readable, and can be extended with new commands.

While SDH has its benefits, it also has its downsides. SDH interfaces or line cards are very expensive, not to mention the equipment switching between these line cards. Another problem is that all SDH equipment is crafted to operate at certain speed. For example, if a carrier wants to upgrade from 2.5 Gbps to 10 Gbps, everything must be replaced. The processing of SDH containers or timeslots is based on the presumption that the traffic being carried is very predictable and uniform by nature. This is totally opposite of internet traffic, which is non-uniform and bursty. Thus SDH is not in itself a viable solution for a network that needs to offer packet-based services. Lastly, SDH is an extremely complex technology both to build and operate, and having to add additional layers on top of it does not really help. [32]

SDH has been the industry transport standard for a very long time, but due to the problems mentioned above in the future it will most likely be phased out in favor of CET. Some of these problems have been addressed in ngSDH, and it will be interesting to see how big of a competitor it will become to CET.

Bottom line: SDH has good carrier grade properties but it is very unoptimal for packet transport and has other properties which leads to high costs.

Asynchronous Transfer Mode

Due to invention of the World Wide Web in the 1990s, a great demand grew for packet-based services. Something was needed in order to get the best value out of the underlying SDH or E/T-carrier infrastructure. These networks could still serve packet switched traffic as an underlying bit-pipe technology, even if switching and routing were done externally. For quite some time the most relevant technology in the field respect to this was Asynchronous Transfer Mode (ATM).

ATM is a connection-oriented protocol whose traffic forwarding is based on the notion of establishing virtual circuits between domain edges. ATM transports fixed-size cells inside these circuits. Because of this, it offers both circuit-switched and packet switched traffic paradigms with several QoS classes from constant to unspecified bit rates. ATM has an extensive toolset of OAM tools built-in, from performance and bit error monitoring to various continuity check and path trace messages. When ATM was introduced, the fastest interface offered 50% faster speeds than Ethernet at the time could, making it a tempting solution for offering several transport services multiplexed into the same backbone. In time ATM was expanded to offer LAN emulation with full IP & Address Resolution Protocol (ARP) support in an attempt to cover everything from WANs to LANs. [32]

Many of the positive properties of ATM indeed allowed carriers to succesfully build packet switched networks scalable from MAN to WAN. In a typical late-90s ATM-driven carrier environment SDH transports the "bulk" of traffic while ATM switches control the traffic flow to and from many kinds of equipment: Mobile Base Station Controllers, IPv4 routers, Private Branch Exchanges, Digital Subscriber Line Access Multiplexers and so on. ATM networks are most often configured with a vendor-specific management solution. When deploying IP services on top of ATM, a carrier can find itself using five or six different type of stovepipe management systems to supply a single service, or spending large amounts of resources to build an in-house system that aggregated all these systems. An example ATM use case is shown in Figure 2.6(b).

Despite its benefits ATM was never adopted as the definite transport service technology. It tried to be a jack of all trades but ended up being master of none. ATM was considered too complex and expensive, which did not in the least ease the complexity of SDH management. Additional hurdles for ATM were the emergence of Gigabit Ethernet interfaces which became a direct (and cheaper) competitor to it both in LAN and access networks. Besides, ATM never succeeded in being a replacement for circuit services; the buffering and the overhead incurred in using a cell header often resulted in undesired delay and jitter when the network got congested, contrary to promises. [28]

Bottom line: ATM failed because it was too expensive, tried to reach every possible networking niche simultaneously and was not able to deliver all features required from a carrier grade service. Carriers gradually opted for cheaper solutions.

Multiprotocol Label Switching

The original idea behind Multiprotocol Label Switching was to speed up L3 routing to level of L2 switching. It adopted many ideas from ATM such as the concept of virtual circuits.

In MPLS, forwarding decision of a core router is changed from IPv4 header lookup against prefix tables to a simple label observation. The label is a small 4-octet field inserted before the payload of a frame, which always corresponds to a specific portion of a path inside the carrier network. Thus with MPLS the egress port for a frame is found in constant time in a core router. At the time this was a precious advancement as router hardware was not fast enough to handle IP prefix lookups at maximum interface speeds. Nowadays this is not a problem as the processing speed of ASIC-based routers has increased, but MPLS is still being adopted eagerly due to its ability to provide packet transport services. [20]

Packet based transport has many advantages. For one, they are more elastic in allocating capacity for bursty packet flows due to frame (instead of timeslot) level statistical multiplexing. Other advantages are that slicing the network into virtual segments is easier, and as an overall the transport network becomes simplified in contrast to circuit based networks. All this eliminates CAPEX and OPEX costs.

Packet based transport frames always contain a label that can be used to assign the packet into a flow. The label can have globally or locally significant meaning. What is "local" is also relative, and can vary in different network technologies, but for MPLS the labels do not have global meaning. If packet transport could be extended to provide global routing, IP routers would become obsolete, further simplifying network management for carriers.

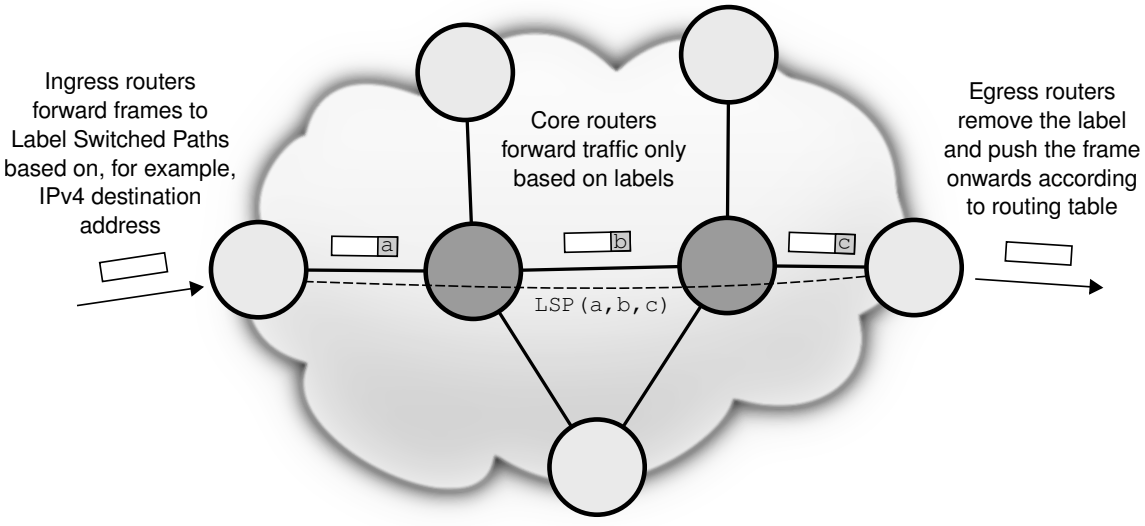


Figure 2.7: An example MPLS core with two core and five label switched edge routers.

In an MPLS network all Label Switched Routers (LSR) residing on network edge form a connection to all other LSRs by so called label switched paths (LSP), as can be seen from Figure 2.7. These connections are persistent by nature, much like virtual circuits. MPLS is thus a connection-oriented technology. However, connectionless services can be built on top of MPLS all the same.

Services are built on top of MPLS by assigning traffic on certain criteria to Forwarding Equivalence Classes (FEC), which in turn can be associated with specific forwarding treatment. For example, when providing a best effort -type IP service, the FEC can dictate that a packet won't be forwarded in the core inside an LSP but with normal (slow) prefix lookup. For a premium IP service, the FEC can associate the packet to an LSR, whereafter the frame is forwarded with fast lookup only based on labels, and is also less prone to be discarded during congestion. In such a scenario the ingress LSR inserts an MPLS shim header under the IPv4 header. The next consecutive LSRs then consult the shim header, possibly switch the label, and forward the packet onwards. The egress LSR removes the shim header and forwards the packet according to routing table. Due to usage of the shim headers MPLS is sometimes refered to as a *Layer 2.5* protocol.

For MPLS to work, internal domain topology must be found with an IGP such as OSPF or IS-IS, after which LSPs are formed. Depending on carrier policy this is either handled automatically with a method that finds shortest paths such as Label Distribution Protocol (LDP), or alternatively done with Traffic Engineering (TE) tools such as Resource Reservation Protocol with Traffic Engineering (RSVP-TE). After this one can attach services to LSPs either by manual configuration or with some kind of MPLS OSS.

While MPLS is an extremely good packet transport technology it is not perfect in every regard. Not all transport service providers run IPv4, and having to use an IPv4-based control plane for controlling MPLS is seen as a weak point by many carriers. The service differentation in MPLS is also too coarse. There are three Traffic Class bits inside the shim header but these do not allow for segregating services inside a single LSP. It has been suggested to use the outermost label for incorporating Differentiated Services in MPLS, but currently there are no vendors providing this functionality. Most importantly, OAM and TE capability is not built in into MPLS. While certain MPLS OAM extensions exist, in production networks LSPs are not usually monitored in any way for performance or health. Because OAM should be pushed to LSPs, which are completely disjoint from the IP-based control plane, it is difficult to develop a completely bomb-proof OSS for controlling MPLS OAM. Therefore, if a carrier wants to enforce strict bounded metrics such as high constant bit rate to a service provided with MPLS, this is usually done by simply overprovisioning the network with as fast interfaces as possible along the route of the service. In the absence of a centralized provisioning system, RSVP-TE becomes extremely challenging to operate in large networks. In such cases, LDP is usually the mechanism used for signalling LSPs, which leads to unoptimal paths and lack of guaranteed link-disjoint protection routes. Restoration must then be achieved on the service—not transport—level, for example routing to a next-best

next hop around the domain in the case of IP traffic. [25, 9]

Bottom line: IP/MPLS-based solutions cannot leverage revenue to carriers beyond "standard" IP and VPN transport services, because MPLS control plane architecture hinders the OAM and TE capabilities required for a carrier grade transport.

Internet Protocol

What about IP, then? After all, it can nowadays be found in almost every location where connectivity services are required, and many programs exist for creating L2VPNs between sites with IP connectivity. Thus in a sense a natural thought would be to offer Carrier Ethernet on top of IP.

In reality this is not an option due to how IP networks are managed. As is the case with MPLS, also in pure IP networks it is usually more cost-effective to over-provision the network compared to running complex and tedious management systems and signalling protocols which offer end-to-end QoS for IP flows. This is one of the reasons why Integrated or Differentiated Services have had much less success than they ought to have: it is simply cheaper to add more powerful routers, more links or faster line cards every time the network gets too congested. This leaves the network underutilized but in the end results in better OPEX for most ISPs than using traffic differentiation mechanisms would yield. Not surprisingly the ability of Ethernet to constantly introduce faster interfaces is also a success factor behind all this. [43, 32]

Another thing is that crafting an SLA for IP networks is troublesome. IP traffic is inherently best effort, thus it is hard to concretely define boundaries for an IP service delivered on top of IP. One suggestion has been to define relative service classes, where a "platinum" class is only guaranteed treatment *better* than a less premium class like "gold", which in turn gets priority over "silver" class and so on. In this way no absolute QoS guarantees are made, which is not very appealing to customers. Besides, we have already established that the less layers a transport network has the better, and IP cannot be transported without an underlying link layer technology. [43]

Bottom line: Despite its ubiquitousness, IP is not suitable for transporting carrier grade services.

2.4 Carrier Ethernet Transport

The general consensus is that whatever the actual technology that Carrier Ethernet Transport is built with is, it should be able to provide both circuit switched and packet switched transport services. Figure 2.8 shows a range of services, all of which will need to be carried on a future transport network. As there are point-to-point and multipoint services in addition to services requiring synchronization,

great flexibility is expected from CET.

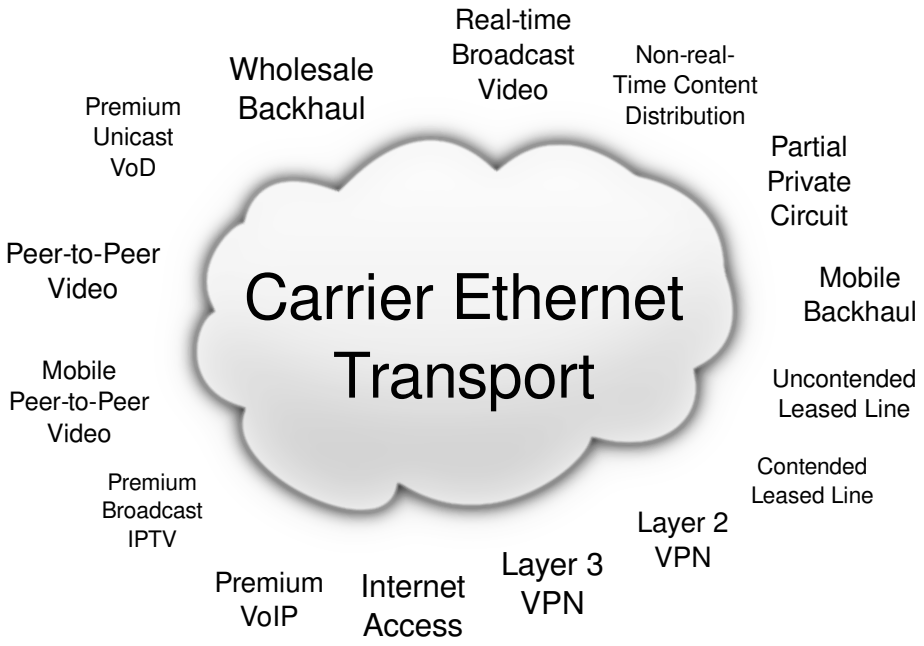


Figure 2.8: Carrier Ethernet Transport -driven transport services. Adapted from [4].

CET is by no means a finished architecture nor standard but more like a concept or an umbrella term. The same goes for management of CET networks; no unified standard exists yet. Therefore a CET management system should be as modular as possible. It should be able to effortlessly operate and interface with different kinds of data planes and possibly their controlling OSS.

2.4.1 Standing on a Crossroads: CET Candidates

Currently there are at least five technologies that try to solve the problem of offering transport services on top of Ethernet. The common denominator is that all support traffic forwarding based on some pre-provisioned transport labeling system instead of forwarding tables propagated with normal dynamic learning & flooding mechanisms and spanning trees. Centralized bookkeeping about services and utilization is the most feasible way of providing carrier grade services with Ethernet. Now we will briefly review these technologies. [3]

Provider Backbone Bridges

Provider Backbone Bridges (PBB) is a technology initially developed by Nortel and later standardized by IEEE as 802.1ah. It leverages total separation of the provider domain from the customer networks.

Table 2.4: Structure of an IEEE 802.1ah (MAC-in-MAC) frame and the I-TAG inside it.

(a) An 802.1ah frame carrying an 802.1ad frame as payload.

	octets
Backbone DA	6
Backbone SA	6
Backbone VLAN	4
I-TAG	6
Destination address	6
Source address	6
Service tag	4
Customer tag	4
EtherType/Size	2
Payload	46—1500
⋮	

(b) Structure of the I-TAG.

	bits
TPID	16
PCP	3
CFI	1
Reserved	12
I-SID	24

An example of a PBB frame can be seen in table 2.4(a). First in the frame are backbone destination and source MAC addresses exactly as in a normal Ethernet frame. To differentiate, they are called B-DA and B-SA. Next up is a backbone VLAN identifier (B-VID). An aggregate of these three fields is used to forward the frames inside the carrier domain. [10]

The next part of the frame contains the I-TAG, which in turn contains the service instance identifier, or I-SID. The contents of the I-TAG can be seen in table 2.4(b). The I-SID is essentially a virtual broadcast domain inside the tunnels provided by B-VIDs. I-SIDs are visible only on edges of a PBB network. Finally the frame contains customer payload. Currently payloads from Q-in-Q to VLAN to Ethernet are standardized, but other payloads such as pseudowire emulation edge-to-edge (PWE3)[2] are also in the making. With current mechanisms customers could be identified—and the I-SID formed—for example based on S-VID, C-VID or a normal Q tag. [10, 37]

The essential part of PBB is that customer topologies, addressing and VLAN spaces are totally separated from the provider domain which transports only provider frame formats with provider administered identifiers. Core PBB bridges must learn only a handful of MAC addresses, even when serving tens of thousands of end sta-

[2]RFC3985

tions. The customer frame is treated only as payload. Customers or customer aggregates, which are represented by the I-SID, are mapped into backbone tunnels, which are represented by the B-VLAN. While this solves many problems, it also creates new ones. Especially transporting Y.1731 and CFM OAM through multiply peered PBB domains is a non-trivial issue. Also, the I-SID address space must be consistent and non-clashing throughout the provider network, which raises requirements for the control plane of a PBB network to keep the I-SID space sane. [10]

Despite the efforts to carry pseudowires and non-Ethernet payload on top of I-SIDs, PBB was essentially developed for scenarios where customer Ethernet networks are connected to PB access networks which are connected to PBB cores, which in turn are peered with other cores. It is not a one-size-fits all solution which can trivally replace IP/MPLS or SDH. Because of this PBB is starting to veer more into a complementary access technology that can be used alongside MPLS networks. [37]

Provider Backbone Bridge Traffic Engineering

IEEE 802.1Qay or Provider Backbone Bridge Traffic Engineering (PBB-TE) is a profile of PBB which retains the PBB frame encapsulation format while discarding most of its other properties. However, it can also co-exist in the same networks as PBB, and can even be used to traffic engineer PBB domains, as the name suggests.

PBB-TE is strongly based on the concept of forwarding traffic via tunnels or Ethernet Switched Paths (ESP) that are preconfigured by a centralized network management system on the management plane. The control plane of PBB-TE remains unestablished. Generalized MPLS (GMPLS), a control plane for physically diverse data planes, has been suggested for this but no actual standardization has occurred. All current PBB-TE implementations rely on static NMS solutions. [3]

In PBB-TE, the 802.1ah backbone MAC addresses (B-DA, B-SA) in conjunction with the backbone VLAN tag are used to explicitly define the egress port(s) for a frame. The concept is simple; with tuples of aforementioned fields it is possible to use the same forwarding mechanism to forward E-LAN, E-tree and E-line traffic. An example of a PBB-TE E-LAN and E-line operating with the same backbone VLAN tag but on disjoint paths can be seen in Figure 2.9

In PBB-TE most traditional Ethernet traits that have still been incorporated in PBB are thrown away. MAC learning, spanning trees, flooding of unknown frames are all absent. Instead the NMS is used to calculate tunnels that are signalled into forwarding tables. This also allows calculation of protection paths and other TE operations. PBB-TE could be used in conjunction with CFM and 802.3ah to provide SDH-like resilience. [3]

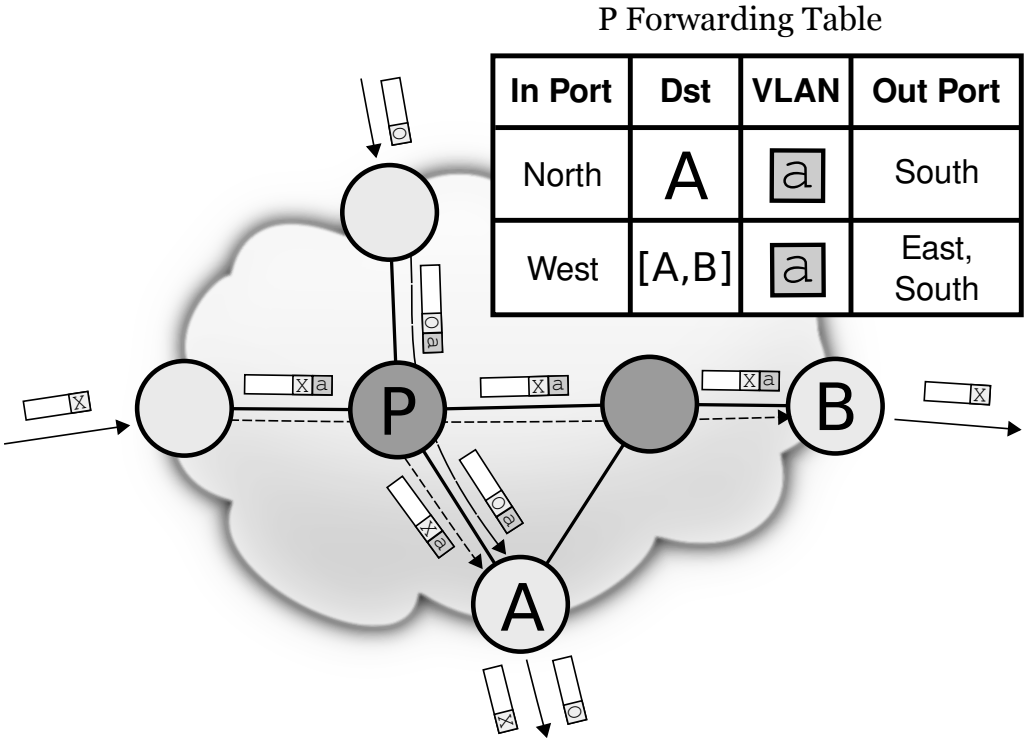


Figure 2.9: Frames being forwarded in a PBB-TE network. Frame information apart from VLAN tags is omitted for simplicity. The arrows inside the domain indicate the frames are being sent to preprovisioned Ethernet Switched Paths with known destinations. This allows the reuse of VLANs, even for paths traveling through the same node. Adapted from [3].

Ethernet Label Switching

Ethernet Label Switching, also known as VLAN Cross Connect (ELS, VLAN XC), is a method for provisioning CET services with Q-in-Q framing. Although work on it was started by IETF and IEEE, it has so far remained unstardardized[3]. The main idea in ELS is that instead of switching based on MAC addresses and VLANs, only the ingress port and a VLAN tag is used to forward frames throughout the network. Framings proposed for this are 802.1Q and 802.1ad. In the latter the aggregated address space of 12+12 bits in Q-in-Q tags is used to achieve 16M VLANs per core port. In ELS terminology the Q-tag is a so called VXC tag and the aggregated Q-in-Q tag so called extended VXC tag. The VXC tags can be swapped and reused inside the domain quite in the same fashion as the MPLS shim header. ELS is by nature transport oriented because the paths formed by the chain of VXC tags are meant to be persistent. An example of an ELS network can be seen in Figure 2.10.

[3]IETF Internet-Drafts Database shows both draft-dimitri-gels-framework-00.txt and draft-sprecher-gels-ethernet-vlan-xc-01 to have an *Expired* status.

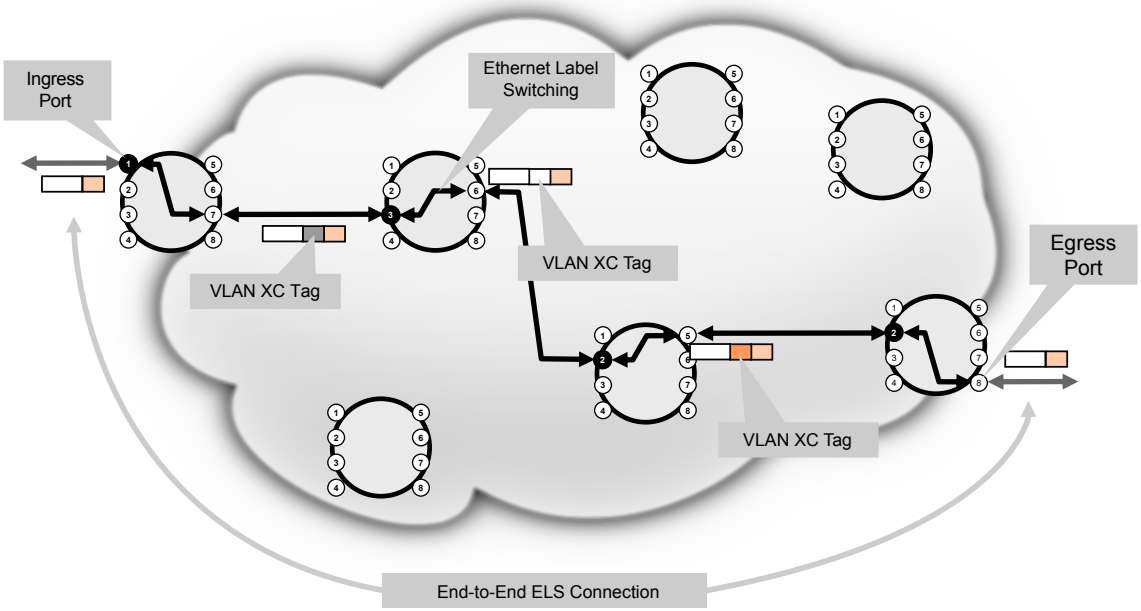


Figure 2.10: An example of Ethernet Label Switching. Adapted from [36].

As for U-NI, ELS supports untagged and Q-tagged frames. Untagged frames are bundled into the same VXC. For tagged frames the customer tag can be either preserved or dropped. ELS forwarding also allows for service multiplexing; multiple customer VLANs in the same port can be associated with a single VXC, or on the other hand specific VLANs in the same port can be directed to separate VXCs. ELS can also co-exist with normal VLANs and provider bridges residing in same hardware, although the VLAN tag space must be partitioned for this. All this imposes additional demands for the system that is controlling the customer VLAN to VXC assignments. [36]

ELS drafts have not endorsed a particular control plane. One of the reasons ELS development has been lagging is that GMPLS standardization for Ethernet has been slow. Capabilities such as traffic engineering, resiliency and multipoint support have been suggested, but for now ELS development seems to have slowed down in the wake of IETF focusing on MPLS Transport Profile. [3]

MPLS Transport Profile

MPLS Transport Profile was started as a joint effort by ITU-T and the IETF in 2008. It is a connection-oriented technology, largely based on the original MPLS specification[4] and emphasizes the utilization of the PWE3 architecture. The objective is to bring MPLS closer to a level where it meets transport and carrier grade requirements and provides reliable transport of any traffic type at the best euro/bit

[4]RFC3031

ratio. Additionally MPLS-TP tries to offer good scalability properties performance monitoring and multi-domain support. This is done by extending MPLS forwarding, OAM, network management and control plane protocols. [1]

An example where MPLS-TP is used in conjunction with Ethernet and IP / MPLS is shown in Figure 2.11.

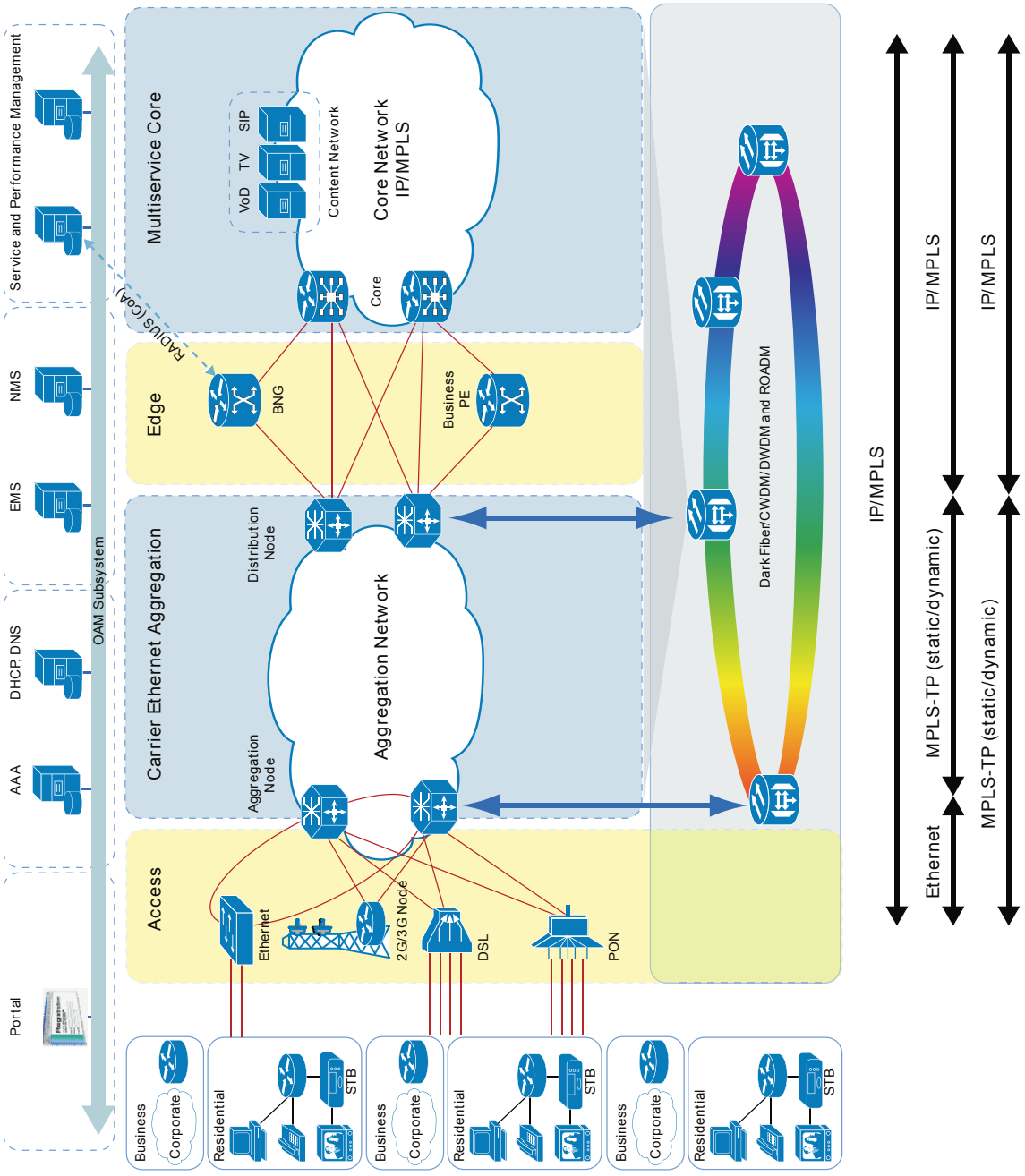


Figure 2.11: Example MPLS-TP deployments alongisde a conventional IP/MPLS network. [31]

To achieve this, MPLS-TP has been defined as agnostic to both traffic and physical layer. To accomodate synchronized L1 traffic three options have been suggested: an overlay synchronization network, a distributed reference clock solution which feeds GPS (or similar signal) from edges to the network, and simply forwarding clock information on top of the packet domain. The first is the most expensive and is not a likely candidate. The second might be a viable option when Synchronized Ethernet is available. For the last solution either Network Time Protocol[5] (NTP) or preferably Precision Time Protocol (PTP) should be used. Considering physical transmission, MPLS-TP can be carried over SDH and WDM networks with exactly the same Generic Framing Procedure code point as MPLS uses. In the same fashion, MPLS-TP can be carried over Ethernet links with a specific Ethertype. [18, 1]

In MPLS-TP partitioning and layering are used to support various client traffic instances from access to core, in point-to-point and multipoint topologies. Additionally, OAM functions have been defined to offer various connection supervisory, maintenance and troubleshooting tools. The use of a control plane in MPLS-TP is optional, and if no control plane is present, GMPLS can be used instead. The general philosophy has been to decouple the control and data planes to the largest extent possible and by default use only the management plane for provisioning services, but if the need arises, a dedicated control plane system can be used to perform Traffic Engineering tasks. Thus MPLS-TP tries to offer a unified management solution over packet, TDM multiplexes and wavelengths but which can still revert to simplified functionality if TE is not needed.

We have just covered a vast ground from home and corporate LANs to Ethernet transport. In the next chapter we will explain the carrier business model and how management standards can be used as an aid in structuring network management.

[5] RFC1305

Chapter 3

Network Management

In this chapter, first an overview of the carrier as a company is given, with operational areas divided into specific functions. After this follows an analysis how these areas relate to network management, especially control plane operations. Based on this we try to figure out which of the existing network management protocols would be the most suitable for a CET control plane northbound interface.

3.1 Network Management From Business Perspective

Most transport service providers are public companies which offer their stocks for sale either to the general public, or some limited audience, thus transferring the ownership of the company. The purpose of any public company is to generate as much money as possible for its shareholders unless otherwise stated in the founding documents. This can be only achieved by minimizing the operational and capital expenditures and maximizing revenue. It can be said that for any service provider, services already sold should be delivered as near the agreed quality as possible and new services should be created where potential demand can be recognized. [34]

A public company can be divided into a *monetary process* and a *real process*. The former is related to flow of money in and out of company bank reserves in the form of investments, and distribution of profits to shareholders, respectively. The latter deals with flow of goods, services and money to and from clients and supply chains. Figure 3.1 shows how we further divide transport network management functionality inside the real process into inventory, configurations, services, staff and administration. [26]

Inventory contains physical communication infrastructure such as routers, switches, converters, repeaters, landlines, fibers, base station antennas and so on.

Configurations dictate how Inventory should function. Configurations encompass everything from creating multipoint EVCs between given edge devices to en-

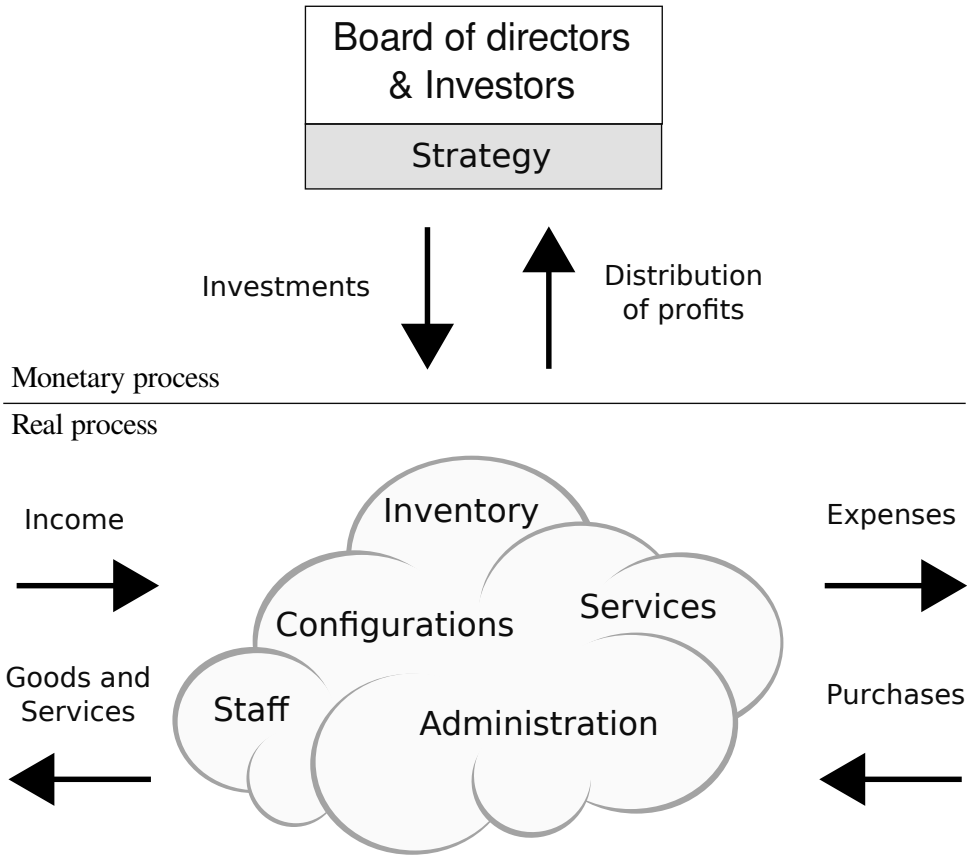


Figure 3.1: Network management in company business context. Adapted from [26].

abling a VLAN in a particular port, to for example setting congestion control thresholds.

Services are where carrier revenue comes from. In this context, a service can be considered any telecommunication capability that is sold to customers. Previously mentioned configurations and inventory are used to realize these services. How an individual frame, cell or a voice circuit is treated in any given situation is dictated by the service that the carrier provides for the customer who is sending or receiving the data.

Administration is needed to handle administrative functions such as payroll management, backing up servers and configurations, keeping inventory software up-to-date, and other miscellaneous, non-network-related matters.

Staff equals the people operating all sections mentioned above. Although we live in a highly computerized world it is impossible to automate everything. A company must have staff in place to perform many kinds of tasks spanning from inventory installation to customer support.

In the monetary process the investors and a board of directors determine the

Strategy that directs the overall business. This can be done in discrete intervals such as the company fiscal year or as a continuous cycle. The decisions they make include, for example, how services will be preferred over one another. Most services in transport networks are carried over a shared physical infrastructure. One cannot fit an infinite number of telephone conversations, high-bandwidth customer VPNs or aggregate ISP transit on top of common packet transport infrastructure without intermittent congestion. Sometimes congestion or faults occurs to the point where services disruptions affect many customers and breach SLAs. Carrier strategy dictates how networks will be designed and provisioned to tackle such problems, and what procedures will take place and which customers are favored over others if such faults do happen and only limited backup resources are available. Of course carrier strategy dictates several other aspects of carrier business as well, from service R&D to creating inter-domain services with fellow carriers.

3.2 Standards and Recommendations

There are several existing standards for network management. This has not always been the case. In the advent of digital telecommunication networks it was often necessary to instantly deploy a new technology when it was available to be competetive. Some of these technologies or systems lacked means of interacting with other systems, forcing the carriers to have separate management processes for each system and effectively creating stovepipes. In the previous chapter we established that this is unoptimal. Only by structured planning and standardized processes can telecommunication networks and their management be efficient. In the following section we will go through some essential standards.

Telecommunication Management Network

The Telecommunications Management Network (TMN) model is a recommendation by ITU-T intended for managing a variety of networks from transport to access [21]. It builds strongly on the Open Systems Interconnection (OSI) reference model, and therefore OSI conformance is required from systems implementing the TMN model. Four management architectures are defined in TMN: functional, physical, informational and logical. The most interesting part of TMN regarding transport network management resides at the logical level. This level defines the following five layers of abstraction:

Business Management Layer (BML) implements policies and strategy of the organization which owns the network. The purpose of BML is to manage business as a whole, see that return on investment is met, company stays profitable and is able to attain wanted market share.

Service Management Layer (SML) manages contractual aspects of services that customers are buying. An important functionality of SML is to generate and

maintain service statistics from raw data, and based on these statistics see that QoS levels are met.

Network Management Layer (NML) is used to manage in some particular scope, network elements that provide services. NML also provide statistics about availability, performance and other such measurements to SML.

Element Management Layer (EML) controls a subset of network elements. It provides gateway/proxy functionality allowing NML to command network elements. Additionally, it logs statistics from network elements below.

Network Element Layer (NEL) is used to configure individual devices like switches or multiplexers.

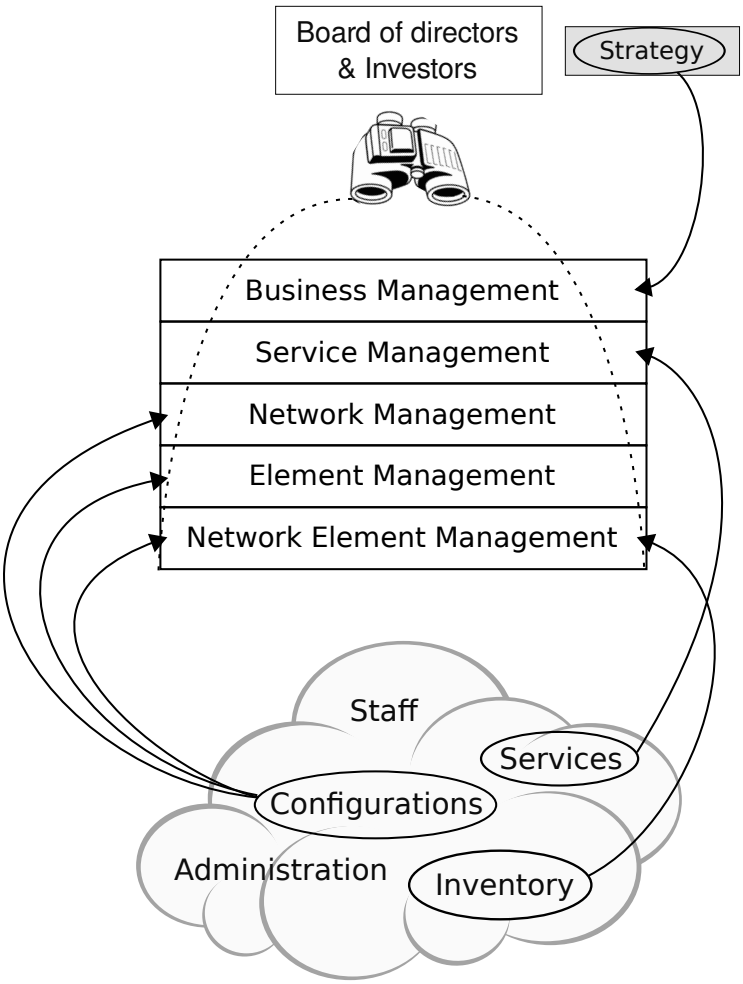


Figure 3.2: Mapping of carrier business objects to the TMN model.

The key benefit of this approach is that it gives an alignment between a company's IT assets and the company's core business. One can say that TMN

gives executive staff the ability to understand the impact to services imposed by the underlying technology. In most cases a customer or the carrier board of directors is not interested in a specific switch of the carrier being up or down or what is the one-way delay on a given path; they are mostly interested if service levels are in acceptable levels. Thus TMN allows inventory and configuration problems that may have impact in business processes to be anticipated better. An example of this is given in Figure 3.2 where a subset of the network management functions discussed in the previous section are mapped to TMN logical architecture layers. [35]

In addition to aforementioned layers the TMN model defines interfaces for exchanging information between heterogeneous networks and computer systems devised of many kinds of technologies. These interfaces, and the data models they operate on, are designed to be able to cope with just about any kind of network management task in existence, and are therefore extremely complex. This is enforced by emphasising multivendor interconnectivity and hierarchy modelling. Everything is based on the notion of exchanging Managed Objects (MO) between management systems. MOs in turn are composed of attributes, methods and behavior. The processing of MOs can be filtered or scoped, and if synchronization is needed for some complex operations, atomicity is supported. All exchanged information cascades as MOs all the way from NEL to BML. [16]

TMN basically forces a structured approach in all facets of network management where one tries to implement it. This can be viewed as a strength because the resulting network management scheme is very modular and can adjust easily to for example addition or removal of hardware, but it can also be viewed as a weakness because it makes TMN hard to implement. The requirement of full OSI implementation at the so-called Q3 interface used for exchanging information between elements can be an obstacle, because a full-blown OSI network from controlling end stations to core switches is expensive to build. Not nearly all LAN, MAN and WAN equipment vendors support the OSI stack. A full-blown TMN network management solution is resource-consuming to develop and operate, which is why only a subset of TMN is implemented in most environments. [27, 16]

Fault, Configuration, Accounting, Performance and Security

ITU-T has also released a subsequent management standard to TMN. It is best known for the Fault, Configuration, Accounting, Performance and Security (FCAPS) concept [22]. This concept was later refined by ISO, who embedded it into the OSI model.

Where TMN is focused on presenting technology in the scope of business, FCAPS is more concentrated in the functional technology management. Actually, it does not try to address business issues at all. In FCAPS, as the name suggests, five different types of information handled by an OSS are described. Some or all of these are handled on different layers of the TMN model. For example accounting is done on several layers; company bookkeeping happens in BML and actual meters

collecting the accounting information reside in EML.

The FCAPS areas are distinguished as follows.

Fault management is about identifying problem situations and limiting them to the exact source of the problem in order to remedy them.

Configuration management concerns mainly configuration of equipment and all adjacent duties such as configuration backups and roll-outs of new configurations. Without configuration management the knowledge of changes made to devices would in the worst case be known only to a single employee, creating a single point of failure.

Accounting management (sometimes Assets) addresses gathering statistic and billing information about sold services.

Performance management is about analyzing long and short-term statistics gathered from the network. It's a valuable tool in planning and provisioning the network or when trying to make sense of repeating trends.

Security management addresses the rights with which users, be they staff or customers, access data and equipment in the carrier domain.

As can be seen, FCAPS is very technically oriented. It is best used in planning real-time processes related to the network itself, such as maximizing the availability of a specific service. When designing and implementing an OSS/NMS with FCAPS model, each component should be taken into consideration; how the OSS manages faults, how data is collected, how configuration changes are rolled out, how accounting is done, and how information breaches are prevented. [35]

Network Management Standards Complement One Another

The standards mentioned so far do not exclude each other; on the contrary, they (and many other network management standards) can be used to complement one another. FCAPS can be used to collect information, TMN provides mechanisms to convert this data into business context. And it need not stop there. An important standard in this regard is the Information Technology Infrastructure Library (ITIL). ITIL is a collection of best practices for management of IT services. These best practices can be utilized to maintain high service quality and overcome difficulties associated with the growth of IT systems. The benefit to carriers from ITIL is that it is in essence a guideline for running an efficient IT organization. Thus, ITIL can streamline the processes of an organization utilizing FCAPS and TMN, but it can also be used to assure that services are manageable throughout their entire lifecycle. [33]

Many other standards exist that could also be utilized. The enhanced Telecom Operations Map (eTOM) is a standard equivalent to ITIL but more focused on telecommunications management. eTOM uses the *customer* as an all-embracing unifier to which everything is eventually connected. The division of underlying

themes is a bit different from the TMN model; operations are kept separate from strategic, infrastructural and product-related processes. Like ITIL, eTOM tries to cover the entire lifespan of services, starting from strategy, moving through design and installation into running the service on a day-to-day basis, charging the customer and finally doing graceful shutdown of the service. eTOM, along with New Generation Operations System and Software (NGOSS), represents the newest in network management. The interesting bit about NGOSS is that it provides a so-called Technology Neutral Architecture (TNA) which describes OSS components in a technology-independent manner. Using for example XML, the TNA can be mapped to a Technology Specific Architecture. While eTOM and NGOSS are too complex to be discussed within the limits of this thesis, anyone looking for the bleeding edge in network management should have a look at them. [35, 23]

Now that we have covered essential network management standards it is time to have a look at what options we have for defining a northbound interface for our control plane.

3.3 Possible Northbound Interfaces

In the following we review the candidates for a northbound interface with which we could manage the ETNA intra-domain control plane. Our target is that the interface should offer the ability to manage the control plane as an equivalent to the Network Management Layer of the TMN model.

TMN Q3

In addition to defining various layers of network management the TMN model defines an interface for conseptualizing functionality and data in a network element and offering methods for accessing this data. The interface is called Q3, and as discussed, full implementation of it requires the presence of a full OSI stack library which makes it an unattractive candidate for the control plane northbound interface. It is quite complex to implement as well.

Simple Network Management Protocol

The IETF-defined Simple Network Management Protocol (SNMP) is often seen as a flipside of TMN; Q3 is used to manage telecommunication networks and SNMP is used to manage IP networks. Since all versions of SNMP rely on UDP/IP connectivity to network inventory it is not a strong candidate for transport network management. However, it is important to mention SNMP simply because it is so popular in todays networks. SNMP has a much poorer information model and almost no hierarchy built-in, but just because it is easy to implement and its agents

require almost no effort to set up due to IP ubiquity, many vendors opt for SNMP as their configuration interface. [27]

Web Services with SOAP

Simple Object Access Procotol (SOAP) along with Web Service Description Language (WSDL) allows quickly creating interface skeletons for accessing a given system. They rely on a simple pragma where all aspects of a system or service are described in a single XML-formatted file. The means for developing web services are thus as follows:

1. Fetch the WSDL file for the service
2. Generate client-side stubs from WSDL file for required language
3. Write client application logic using stubs
4. Make client application invoke the service.

The process is straightforward and quite similar to writing Common Object Resource Broker Architecture (CORBA) applications. The strength here is that client and server software is decoupled in a very strict manner. One can use a totally different operating system, development kit, hardware etc. in the server than in the client. In addition, due to the object-oriented nature of SOAP, one can make changes to the WSDL without necessarily breaking the client applications at all. [43]

What this means for transport networks is that the nomenclature, functionality, topology and various aspects of a carrier domain can be encapsulated into a single file and provided for a third party, should the need arise for developing external software for accessing a particular transport network system.

CORBA

CORBA is a standard that allows software written in various languages and running on several computers to work together. Much like SOAP/WSDL, it relies on creating stub/skeleton methods and populating them during implementation. The transparent definition of a service is done with a so called Interface Definition Language (IDL). The IDL specification is then mapped to some specific implementation language, such as C++ or Python.

Operational Support Systems through Java

Operational Support Systems through Java (OSS/J) is a project which aims to develop Application Programming Interfaces allowing integration of Business and

Operational Support Systems. From our viewpoint OSS/J is not an extremely suitable solution due to additional overhead incurred by the various business functionalities.

Remote Console

One option we considered for the northbound interface would be to access the system with telnet or SSH and use the local shell of the management system to send control messages to management applications. The compelling thing in this is that it would allow extremely fast prototyping. On the other hand, the interface would then rely of a specific shell and possibly a specific operating system.

After some research into the subject, we selected SOAP from these candidates. It allows describing generic functionality that is beneficial in both FCAPS and TMN type of information gathering. It also has excellent support in the form of commercial and open source tools for many development platforms and allows for speedy definition of an interface skeleton (and updating the interface later on). Additionally, with SOAP it is possible to write clients for any platform from a web browser to standalone software running in an embedded system.

Now that we have covered some essentials about the transport service provider organization and relevant management standards, we can move on to the actual ETNA intra-domain control plane.

Chapter 4

ETNA Intra-domain Control Plane

This chapter explains how ETNA tries to address the issues of delivering Carrier Ethernet services on control plane level. A central item in this solution is the network management system developed for the 7th framework ETNA project by the author.

First we discuss what sort of components the ETNA architecture comprises of and how the system, from an architectural standpoint, tries to fullfill CET requirements. Then we explain how ETNA management, control and data planes are constructed and interface each other. After this we delve into details of the actual ETNA control plane management system.

4.1 General ETNA Architecture

The general vision in ETNA has been to separate three layers of functionality in the network: transport layer, transport service layer and network service layer.

Transport Layer (TL) offers L2 connectivity primitives in the form of point-to-point tunnels or point-to-multipoint trees. In TL a tunnel or a tree is nothing more than a path or a set of paths traversing through the provider network which translate into simple data plane forwarding entries: a frame with specific backbone destination address and backbone VLAN tag is forwarded into a specific port.

Transport Service Layer (TSL) is where actual services are built on top of existing TL primitives. In transport service layer criteria such as minimum delay, jitter, capacity constraints and protection demand are defined and attached to tunnels as they are created. The aim is to create a flexible service platform which is capable of adjusting to many types of services and also allows the creation of new service types when the need arises. For example, TSL should

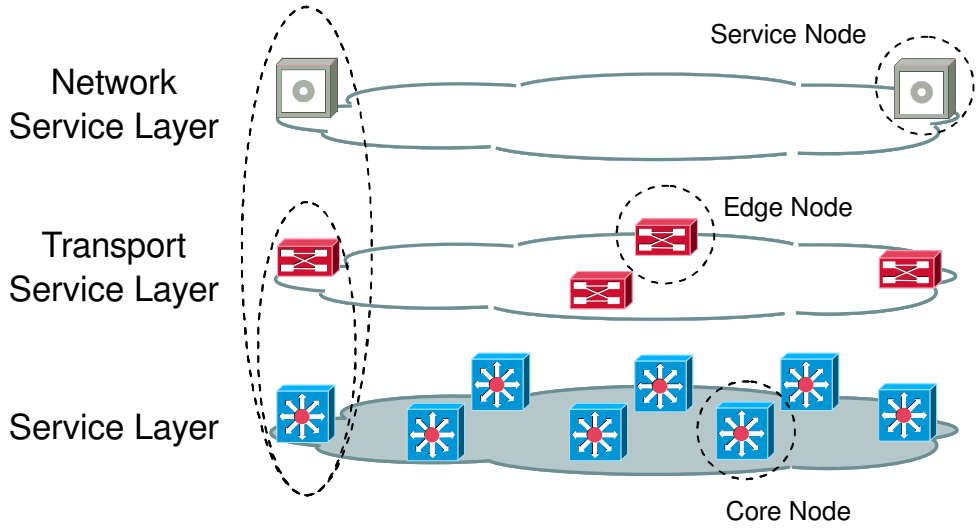


Figure 4.1: ETNA three-layer architecture. [4]

be able to transport MEF-defined services from E-Line to E-LAN, with full MEF OAM compatibility on the provider edge for interoperating with other carrier environments.

Network Service Layer (NSL) adds a level of control on top of transport service layer. Here we can provide value-added services which relate to, say, the payload of customer traffic. Instead of simply forwarding based on L2 information we can run an IP routing service on top of the transport infrastructure. In this IP-over-ETNA model each network service -enabled edge device would run IP routing processes and push frames into correct TSL tunnels much in the same way MPLS is used today. Yet another example is various mobility services—they require MAC learning which is not available on TL or TSL.

In figure 4.2 the "fat" pipes between nodes are TL tunnels. The thinner paths inside these tunnels represent TSL connections, where particular U-NIs are linked to TL tunnels to form a service. Two of the nodes also contain NSL functionality, depicted by a router. This could mean that if the node would receive frames with a specific Ethertype in a specific port, the frames would be redirected into a NSL processing engine. If the frames contained IPv4 traffic and the NSL engine would have IPv4 routing enabled, the frames could be forwarded into a particular TL or TSL tunnel based on the IPv4 header instead of VLAN tag.

While the three-layer architecture is the "normative" specification, it is far from the total potential of ETNA. From network services it is no more a long stretch into a **Virtualization Service Layer (VSL)**. With VSL, the underlying network infrastructure can be totally masked out from third parties. This in turn would allow transport service providers to open up and provision parts of their networks to service providers operating across multiple carriers.

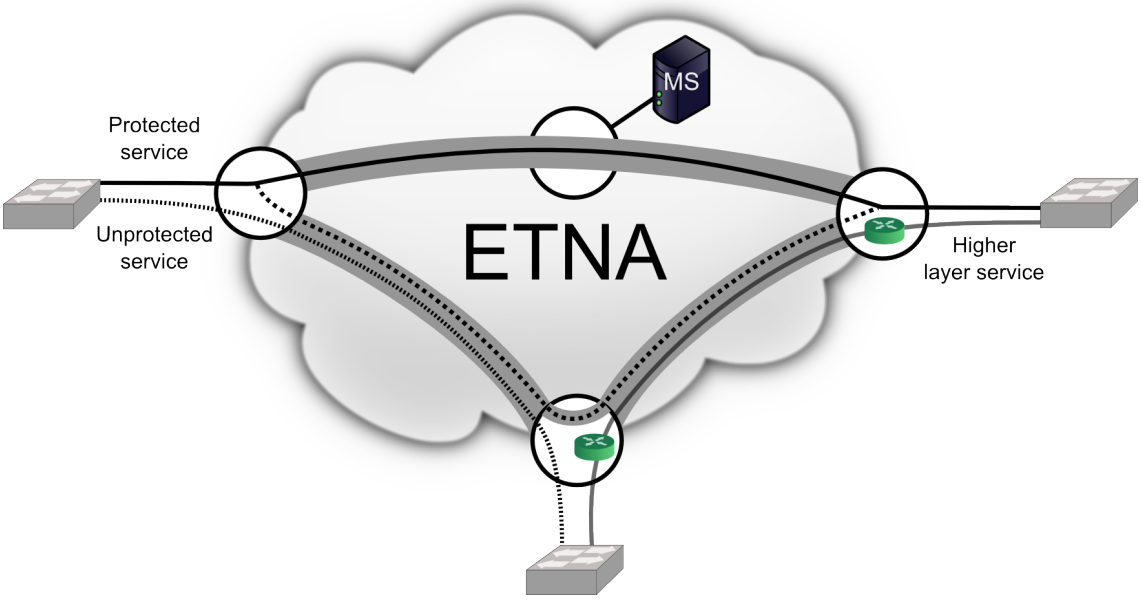


Figure 4.2: Services on top of ETNA. Wide tunnels represent Transport Layer, thin end-to-end tunnels between customer switches represent Tranport Service Layer and two virtual routers represent Network Service Layer.

Put another way, consider a carrier which is operating on all of these 3+1 ETNA layers. A particular carrier could reserve for example 20% of its overall capacity for administrative purposes and mission critical traffic. 30% would be latched for customer VPNs, 40% for best effort Internet and 10% would be offered on an auction basis. If a particular service provider has a sudden need of capacity, the carrier could in essence sell a virtual transport domain to service providers from its own infrastructure. Service providers could then operate virtual service nodes in provider nodes to deliver their services. As carriers are unwilling to expose their network infrastructure to third parties they could let the virtual domains be edge-constrained so that service providers would only get information about edge nodes, which other edge nodes they are connected to, and which external operators they are connected to outside the domain. The core infrastructure would be totally transparent. A virtual service node might ask for example for a two-hour reservation of constant 2 Mbit/s traffic from itself to another edge node. The transport service layer would then see if this is possible and respond accordingly with either success or failure. Some of this functionality could already be done with MEF 16.

Now imagine a pan-European or even global traffic engineered CET network where multiple such transport providers are interconnected in all possible countries in the world. A service provider is now capable of offering any kind of service, be it framed traffic from Ethernet to IP to cell/circuit switched traffic to “light switched” WDM wavelengths. Everything is provisioned on-demand, and every service delivery aspect from SLA parameters to mobility to charging is handled automatically by

the internetworked cluster of ETNA domains. Also, service providers may have preprovisioned sets of SLA templates. If paramers in pre-provisioned SLAs do not align, a broker system could be used to combine them to nearest-possible equivalents and offer an aggregate SLA. Or the broker could even suggest adding new template into each affected system for this particular case.

This is the real vision of ETNA. Being able to use any service on demand, on any network, on any device, on any physical medium and being able to seamlessly roam with that service to any other network, device or physical medium. And doing all this while maintaining security, usability, Quality of Service, and requiring only minimum modifications to user equipment.

4.2 Data, Control and Management Planes

In ETNA the control, data and management planes are separated logically and physically. However, they are perpendicular to the layered architecture described in the previous section. When a given TSL or NSL service is provisioned, processing is done on all three planes. Management plane can have capability to negotiate about SLA parameters for a given EVC, and the composition of these parameters is enforced in the data plane shaper and forwarder modules when the service is rolled out.

Management plane can be considered as an intermediary between clients of the transport service provider and the control plane. Management plane commands the control plane on where to install tunnels and what kind. All the service profiles are inserted to the control plane from the management plane, as well as information on specific customers, their points of presence, and services which are sold to them.

Control plane consists of distributed Control Elements (CE) and a centralized Management System (MS). The rationale behind this is that distributed path computation schemes always leave some parts of the network unused as they conclude the shortest paths in a given area. If we want to route traffic intelligently into unused parts of a network as a congestion avoiding mechanism, and on the other hand calculate protection paths (with possibly different path computation algorithms per service), the best solution is to realize routing as a centralized functionality. This is emphasized with multipoint services, where the calculation consumes large amounts of CPU resources the larger the network and the bigger the destination set is. Running the interior gateway protocol which discovers and monitors the topology of the network is an important task for the decentralized part of the control plane, as well as setup, maintenance and teardown of tunnels by signalling. Because the pool of CEs constantly report to the MS about topology and capacity reservations, the MS can calculate requested tunnels against this information in the Path

Computation Element to achieve optimal network utilization. If the default algorithm fails to produce a viable result for the request, the architecture allows a fall-through to other algorithms in order to find a solution.

Data plane is constructed from hardware which should be capable of QoS-constrained forwarding, generating OAM probes between MEPs in millisecond-order intervals, and other functionalities required by Carrier Ethernet. The ETNA data plane consists of Generic Forwarding Elements (GFE). A GFE is a flexible network system which is able to dynamically change its forwarding logics, classification and shaping rules, frame encapsulation methodology, adherence to QoS requirements and various other functionality. [13]

Together, the CE and GFE (or just FE) create a Network Element (NE). CE, FE and NE are terms are from the Forwarding and Control Element Separation (ForCES) protocol framework which is used to form the adjacency between CE and FE. Specifics about the control plane ForCES architecture and its implementation can be read in [14, 6, 7]. An example topology showing CEs, FEs and the MS can be seen in Figure 4.3.

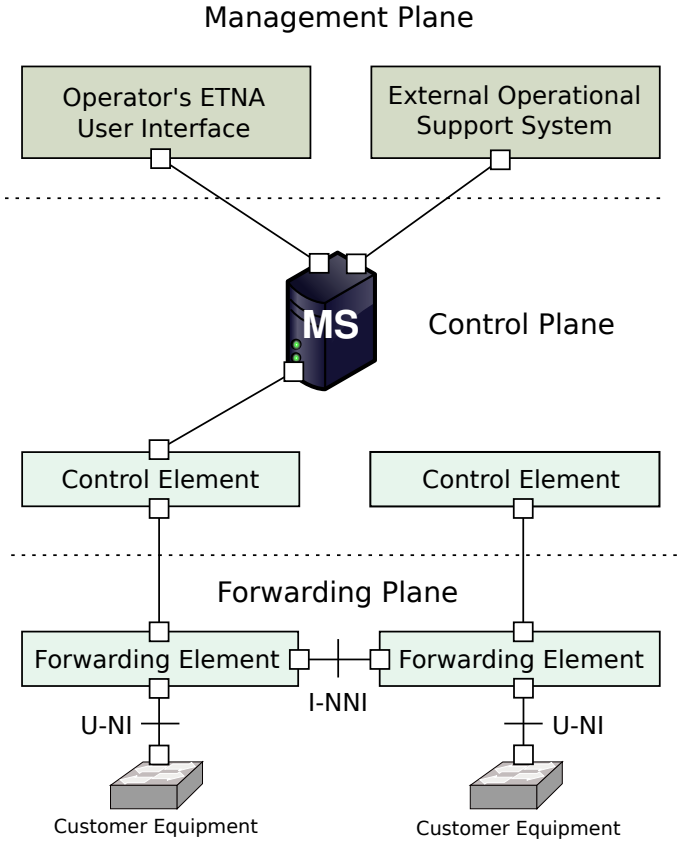


Figure 4.3: Example configuration of physical interfaces (denoted by white boxes) in an ETNA domain.

To obtain characteristics of the underlying data plane the CE queries the FE about its capabilities. As a result the CE knows what kind of frames, with what sort of priorization and at what speed can be sent via the FE's interfaces. CE then configures FE to its liking. After this CE begins sending routing advertisements transparently through FE's interfaces. The FE and CE are described as one and same NE in the link state database of the control plane routing protocol although physically they are separate devices. The CEs are implemented with (virtual) machines that have a Central Processing Unit (CPU) required to keep up with complex signalling and routing tasks, while the FEs are implemented with network processors. A network processor, as opposed to traditional ASIC-based routing and switching hardware, is a device whose forwarding engines can be modified while the device is up and running. This allows every aspect of a transport service to be modified dynamically from framing descriptions to forwarding behavior, without any modication to the FE binaries running on the network processors. This is possible because all necessary framing information is transfered in XML descriptions from CE to FE over the ForCES boundary. The FE XML parser converts these descriptions into rules for a microcode program controlling the forwarding process inside the NP. [13, 12, 45]

In ETNA the CE-FE mapping is always 1:1. If some capabilities are unavailable (say, if an MPLS forwarding plane lacking certain OAM features is attached to the ETNA control plane), the CEs send routing advertisements describing limited functionality. In turn, the MS interprets these routing advertisements by offering constrained services to its northbound interface.

A common caveat against centralizing functionality is that the centralized system forms a single point of failure. This is not a problem in our model since the decentralized cloud of CEs does not care at which I-NNI ports and how many Management Systems are connected to it. Neither does the MS architecture; its functionality can be replicated in several points in the network, the only cumbersome issue being that these systems must then be interfaced separately.

4.3 Control Plane Management System

The purpose of the control plane Management System (MS) is to monitor topology of the network, instruct CEs, react to changes or disruption of services in the network, and provision any services that are requested by the management plane. Because the MS must perform various tasks over various interfaces its architecture was split into modular units instead of a monolithic approach, as can be seen in Figure 4.4.

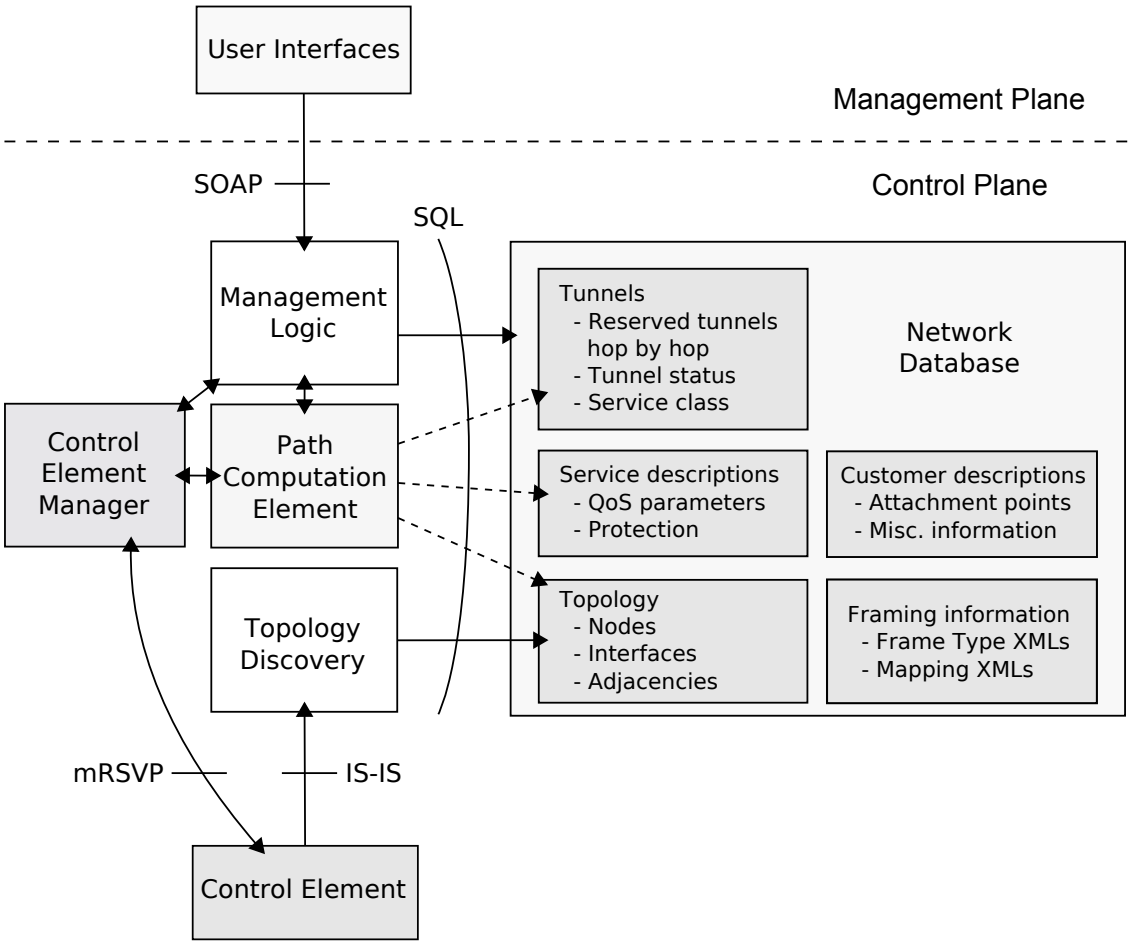


Figure 4.4: Architecture and interfaces of the ETNA Control Plane Management System. Solid arrows denote reads and/or writes, dashed arrows are read only operations. Management Logic operates on all tables of the Network DB, thus its table-specific reads and writes are omitted.

4.3.1 Architecture

The main architecture of the control plane MS consists of a Management Logic (ML) that is the processor of incoming management requests, Control Element Manager (CEM) which manages the decentralized part of the control plane, Network Database (DB) that is a central information and state repository, Topology Discovery (TD) which monitors the relationships between CEs and the Path Computation Element (PCE) that is responsible for route calculation.

Topology collection and maintenance is handled in TD and stored into Network DB. Giving orders to control elements is the task of CEM. Service disruptions and faults are also handled in CEM and information about them is stored into the Network DB. Service provisioning is a task concerning ML, PCE, Network DB and CEM. Figure 4.4 also depicts the northbound (SOAP) and southbound (mRSVP,

IS-IS) interfaces used for instructing the MS, managing the underlying Control Elements and routing inside the domain, respectively. The Network DB can either reside on a single location in the intra-domain architecture, or as a separate DB inside each of the Management Systems. This latter scenario imposes a requirement that either the DBs are synchronized or that each CE broadcasts their available resources in the advertisements of the routing protocol.

4.3.2 Implementation

In this section we will discuss the implementation of the Management System. Apart from the PCE and core Quagga functionality inside TD, all modules were designed and coded by the author. Information for compiling this software as well as running it can be found from Appendix A. The currently implemented software is able to perform ETNA TL and TSL tasks; functionality for managing NSL Service Nodes should be implemented in further research efforts.

Management Logic

ML is the ordering system of the MS, which receives orders from Management Plane via its northbound interface described with WSDL. All controlling WSDL methods are invoked with SOAP.

Arguments required for ML to function correctly are described in Appendix A. After arguments have been parsed and necessary data structures initialized, ML initiates its listen sockets.

Connections are not handled with normal POSIX calls but with functions from the gSOAP toolkit[1] instead. These functions, such as `soap_bind()` used for binding to a socket or `soap_accept()` used for accepting a request, act as wrappers to POSIX functions, masking out the need to perform reads and writes on sockets in the main SOAP handler function. After invoking the previous calls the program has a listening socket ready to process incoming SOAP requests. When a request arrives, function `soap_serve()` delegates the responsibility of handling the request to the linked function object which corresponds to that particular request. If the program has been started in forking mode, it creates a new process to serve each request so the main program can continue listening for more connections. Some requests result in signalling toward CEs while some can be processed merely by querying the Network DB.

After the request has been processed either a response or a fault is returned. A good example of the overall process of handling a SOAP request is the `CreateP2PTunnel()` function call. It is described in the WSDL service description as per Listing 1. The purpose of this function is to create a tunnel in the network. When it is invoked sanity checks are done on all the identifiers to guarantee that

[1] http://gsoap2.sourceforge.net/

Listing 1 Declaration of the point-to-point tunnel creation SOAP function.

```
int etna__CreateP2PTunnel(  bool TestOnly,
                            unsigned int TunnelID,
                            unsigned int ServiceID,
                            long long GuaranteedCapacity,
                            long long MaxCapacity,
                            long long GuaranteedReverseCapacity,
                            long long MaxReverseCapacity,
                            unsigned int SourceNodeID,
                            unsigned int DestinationNodeID,
                            std::string Name,
                            unsigned int TunnelTag,
                            etna__CreateP2PTunnelResponse &return_);
```

for example service class or endpoints for this tunnel exist. After this, a request is sent to the PCE to calculate the optimal path for this tunnel with the requested service class and capacity reservations. If provisioning of the tunnel is possible, the PCE returns a non-null result. As the PCE implementation was done in Python, wrapper classes were built for storing the PCE responses. Classes exist for describing point-to-point, point-to-multipoint and multipoint-to-multipoint paths or path sets. Transformation from Python objects to C++ is done by the Boost library [2]. When ML has received a non-null path computation result it tries to signal the path into the network via the southbound interface of the MS. The sender of the original SOAP request is notified of successful tunnel installation with an OK, or with a Fault message when an error has occurred. All SOAP transactions processed by the ML are ended with either of these responses.

Network Database

The purpose of the Network Database is to support the MS in maintaining network topology and on the other hand keep track of services that have been provisioned into the network.

SQL was chosen as the Network DB engine due to its wide API support in many programming languages, as at the time of designing networks database structure the programming language used to implement other MS software was still undecided. As can be seen from Figure 4.4, the database interfaces allow accessing topology-, customer-, service-, tunnel- and framing-related tables. These are just the logical table groupings; the actual tables, along with the indexes or keys which are used to link the tables together are shown in Figure 4.5.

Node table contains information about Network Elements.

[2] http://www.boost.org/

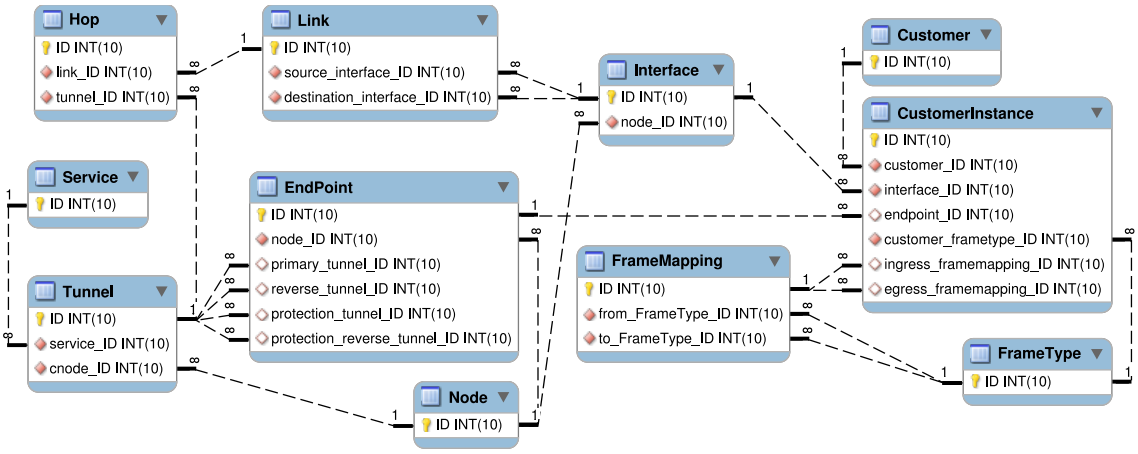


Figure 4.5: Internal structure of the Management System Database.

Interface table contains attributes of the interfaces through which these nodes are capable of forwarding traffic.

Link table is used to describe unidirectional links between interfaces, thus a physical connection is always described by two entries in the link table. These first three tables are populated by Topology Discovery.

Tunnel table gives overall information about tunnels, from service class to reserved capacity.

Hop table lists all the links that the tunnel traverses. ETNA architecture does not support a tunnel traversing the same link twice, thus the ordering of hops inside the table is irrelevant. As a hop is directly linked to a corresponding link entry in the link table point-to-point, point-to-multipoint and multipoint-to-multipoint routes can be described in a single hop table. This makes ETNA a reasonable candidate for a PBB-TE control plane.

Service table describes service envelopes that can be applied to one or more tunnels. When the envelopes are bound to a specific tunnel or multipoint tree that tunnel must respect the parameters of that service. Such parameters can be queuing priorities, protection scheme, maximum delay, or jitter et cetera.

Customer table mainly houses customer reference information, which currently consists of customer ID and name, but can be easily extended to contain charging, billing and miscellaneous information.

Customer instance table has information about domain edge interfaces which contain capacity reservations for customers. In addition it denotes what sort of frames and tags can be expected on the ingress (or should be written to the egress) of these interfaces, again something which is initially queried from the FE by the CE.

Frame type table has information of the frame types that the domain is currently capable of forwarding. ETNA control plane has currently implemented support for normal, 802.1Q, 802.1ad and 802.1ah frame types in addition to an ETNA-custom frame type used in intra-domain forwarding. The Frame Type also indicates whether or not a frame should be able to yield multiple results in a forwarding entry (for multipoint support). As long as the data plane is running on the GFE solution created by BGU, adding new frame types is only a matter of adding new XML descriptions into the system.

Frame mapping table contains mappings from a specific frame type to another, including how standalone customer frame types are mapped to ones where they are encapsulated inside core frames.

The library used to interface the MS DB is MySQL++[3]. With modifications, other libraries could be used as well if other database engines are in place.

Topology Discovery

Monitoring the topology of the domain is the responsibility of the TD unit located both in MS and CEs. There are various approaches of finding out the topology of a given autonomous system with a so called Interior Gateway Protocol (IGP). Popular IGPs operating on network layer are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF). There is also an IGP operating on link layer called Intermediate System to Intermediate System (IS-IS). This protocol was a very natural selection for the base of TD implementation as it is agnostic to the network addresses which it routes. While IS-IS implementations have been equipped to carry IPv4 subnet reachability information, this is not a requirement for the base protocol to operate. The usage of Type-Length-Value (TLV) fields inside the protocol messages allow for easy extendability even with older IS-IS routing hardware, because unknown TLV types are flooded instead of discarding them. Researchers at ComNet have specified new Capability and Interface Parameter TLVs for propagating information that the Management System requires for provisioning packet transport services. This includes per-link capacity reservations, VLAN ranges and exclusion lists, link delay and expense information in addition to various other metrics. Since this information is propagated all the way to the Network DB, it can be utilized when calculating paths in the PCE. [42]

Topology Discovery unit was built on top of the Quagga[4] routing suite by extending it to use the aforementioned TLVs. In order to make the MS conscious of the intra-domain topology, a normal IS-IS adjacency is formed between the MS and the Control Element(s) it is directly connected to. After this, the TD can see all link state packets (LSP) that are advertised within the domain. Monitoring is done in semi-real time, because IS-IS is meant to propagate information in the order of tens

[3] `http://tangentsoft.net/mysql++/`

[4] `http://www.quagga.net/`

to hundreds of seconds. This means that in the control plane architecture TD is only a tool for maintaining topology image, and not suitable for OAM recovery purposes. The topology is inserted into the Network DB with a program that periodically parses the Link State Database (LSDB) of Quagga and compares the current state of the network to the one recorded in the Network DB.

Path Computation Element

Provisioning traffic paths for services is a task where the PCE is a central item. A centralized path computation element is in a sense the main justification of centralizing control plane functionality in Carrier Ethernet Transport environments. As has been stated, dynamic route calculation most likely leads to suboptimal network utilization. Relying on paths formed by spanning trees leaves some links completely unused. Besides, switching hardware is usually not well suited for long-standing path computation. A fully external, centralized entity which has a complete image of network topology and a general purpose Central Processing Unit is far better in this regard. Application Specific Integrated Circuit (ASIC), specialized hardware commonly used in routers and switches, is much more limited in its capabilities of performing memory-consuming path calculations than a CPU.

When the PCE gets a request from the ML to calculate a tunnel or a multipoint tree between several endpoints, it consults the DB to form a graph of the network. Based on this graph and the current capacity reservations (and the requested service level), the PCE computes the optimal paths and returns them to ML. If the service is protected, it also calculates a link-disjoint protection path between the endpoints. The calculation script is external to the Management Logic program, which allows a new algorithm to be inserted on the fly. Currently naïve shortest path Dijkstra and a version based on Bhandari algorithm are implemented. [2]

Control Element Manager

One purpose of the MS is to react to disruptions in service. This is the duty of Control Element Manager (CEM) which listens to alert and notification messages from Control Elements. Currently CEM uses the same management RSVP library that ML uses to contact CEs. CEM is a standalone unit and because it does not deal with any SOAP transactions the MS can still continue to function normally should the management interface die for some reason. CEM is useful for example when a link is broken and a message is sent to inform the MS about the situation. A carrier may have a policy that a *second* backup path is calculated when the primary path fails and traffic is switched to secondary path. In this situation CEM could ask PCE to provision the secondary backup path. CEM would then order the CEs to install a tunnel corresponding to this new path.

4.3.3 Interfaces

Logical connectivity to MS consists of northbound management interface and southbound signalling interfaces. Physical interfaces can be arranged into various configurations. An example setup of physical interfaces in an ETNA intra-domain scenario was seen in Figure 4.3. To form a simple ETNA domain, at bare minimum one MS, one CE and one FE are required.

Northbound interface

The MS provides a northbound interface to the management plane which is described with WSDL and used with SOAP. The service description of ETNA intra-domain control plane, also created by the author, conveys all functionality of the system, independent of any programming language. The methods in the WSDL are categorized into groups related to customers, framing, services, topology and tunnels.

Customer methods can be used to add, remove and edit customer profiles. Additionally, Customer Instances (CI) in ETNA intra-domain edge interfaces can be managed with these methods. CIs specify what kind of frames are used and what sort of service template should be assigned to traffic in a given customer port.

Framing methods are used to manage frame types and mappings supported in the domain if a Generic Forwarding Element -driven data plane is present.

Service methods are used for creating and managing service templates whose properties will be taken into account when provisioning tunnels and CIs.

Topology methods allow performing various queries about topology. They allow both read and write access to topology information, which means it would be possible to construct a Topology Discovery module that would populate topology tables in Network DB over SOAP rather than SQL.

Tunnel methods are used to install, remove, query information about or monitor tunnels inside the domain.

This follows the rationale that several things in a transport network are always present, and these groupings try to reflect these static entities. There will always be network elements forwarding the traffic, so it is natural to specify methods that manage these elements and their interfaces. There are always customers that are sources and sinks for the traffic, and so forth.

However, some things are not specified in the parameters of the functions, such as details about frame types. Instead, the frame-related methods allow uploading XML descriptions of the frames into the system, making the frame type's

internal structure transparent to the MS. The methods specify only a label for a frame type, otherwise the SOAP interface does not dictate how the frame type is described inside the XML. Once received, the descriptions are replicated from MS to CEs as-is, and CEs parse the relevant information out of them for insertion to FEs.

Table 4.1: Most relevant SOAP methods required for provisioning a service with ETNA intra-domain Control Plane.

Type	**Method**	**Description**
Framing	AddFrameType()	Add an XML description of a frame type into the control plane.
Framing	AddFrameMapping()	Add an XML description of a mapping between frame types into the control plane.
Customer	CreateCustomer()	Create a customer profile.
Customer	CreateCustomerInstance()	Signal that a particular interface of a selected node is enabled for assigning customer traffic with selected characteristics into a tunnel.
Customer	MapCIToTunnels()	Activate forwarding entries which forward traffic between a Customer Instance and a tunnel in the ETNA core.
Service	CreateService()	Create a service template describing delay, jitter and other QoS metrics.
Topology	ListFullTopology()	Retrieve a listing of the domain topology, complete with U-NI and E-NNI information.
Topology	SetNodeState()	Set a node as active in path computation.
Tunnel	CreateP2PTunnel()	Create a point-to-point tunnel.
Tunnel	CreateP2MPTunnel()	Create a point-to-multipoint tunnel.

Only a small subset of the 60 methods defined in the WSDL is required to provision, for example, an E-line in an ETNA domain. An example of such a subset is given in table 4.1. The remaining methods can be used to obtain statistics and other information, and on the other hand manage various aspects of the network elements from activating interfaces to attaching new destinations to a multipoint tunnel.

So far three implementations using the MS northbound interface exist. Our team at Comnet developed a Web GUI for assisting development tasks, and a Python GUI for the ETNA demonstrator. A third implementation was developed at the Ben Gurion University (BGU) for the ETNA inter-domain demonstration.

The Web GUI requires a PHP-enabled web server to be able to interface the MS. This GUI is allows delivering MEF-defined E-lines and E-trees, and implements 80% of the functionality defined in the WSDL. An excerpt of the Web UI can be seen in Figure B.1 of Appendix B.

The Python GUI does not use entirely the same subset of WSDL functions as the Web GUI. With the Python GUI it is possible to create an arbitrary test domain consisting of CEs laid out in a 2-dimensional topology. The Python GUI can also instruct the MS to signal services into this test domain and even visualize the signalling process with detailed information about per-hop forwarding entries. A screenshot of the Python interface can be seen in Figure B.2 of Appendix B.

Third northbound implementation is an OSS system completely external to the control plane developed at BGU. In ETNA architecture this OSS is connected to the MS and also to a U-NI port to be able to perform adjacent domain discovery. It bootstraps by sending a request to the MS to signal E-lines with a dedicated inter-domain management VLAN tag between each U-NI. Once this has been completed, the OSS can broadcast inter-domain reachability advertisements to other similar OSSs. Because ETNA inter-domain architecture is disjoint from intra-domain control plane architecture, it does not make sense to duplicate this functionality on the control plane. Therefore the WSDL describing the MS northbound interface does not contain methods or functionality hardwired specifically into inter-domain activities. In current ETNA intra-domain control plane, the very concept of a "domain" exists only in the Area ID of IS-IS Network Service Access Point (NSAP) address. However, control plane architecture does not limit addressing. Any kind of hierarchical domain information can be put into NSAP address of each switch. The ETNA team at Aalto has suggested a format resembling ISO Data Country Code but which fits into a 6-octet MAC address. [5]

Southbound interface

Currently the control plane Management System uses two southbound interfaces: one for Topology Discovery and one for CE management. TD requires a physical L2 interface (such as Ethernet) supported by IS-IS as its southbound interface to be able to populate domain information into Network DB.

For the purposes of managing CEs we developed a protocol called *management RSVP* (mRSVP). mRSVP has the same kind of frame type, header and TLV objects as in traditional RSVP, but additionally contains TLVs dedicated for CET service management, such as tunnel installation or fault indication messages. The latter can be used to convey fault information from CE to MS once an OAM instance has discovered that a given tunnel is down. The choice to craft a custom protocol for intra-domain management instead of, for example, SNMP, was natural since existing SNMP software has a strong presumption that IP connectivity is available. However, the MS is not constrained to using mRSVP. If an ETNA domain is built with CEs that do have IP addresses, it is prefectly possible to replace mRSVP with SNMP.

In the Management Logic section we discussed that after ML has received the result of a path computation from PCE, it begins to signal this result to CEs. Figures C.1 and C.2 in Appendix C give a more precise image of this process. After the first CE receives a mRSVP path installation instruction the CE proceeds to signal the tunnel with a suitable protocol. The MS does not dictate this inter-CE signalling protocol; it can be anything as long as it is able to understand the multipoint, protection and other tunnel information carried inside mRSVP messages. Specifics about our signalling implementation can be read in [25, 41].

In the next chapter we will analyze the ETNA intra-domain control plane introduced in this chapter.

Chapter 5

Evaluation

Now that we have presented the essential information about the ETNA architecture, it is possible to evaluate some properties of both the control plane and the Management System it uses. The first part in this chapter consists of an architectural evaluation about ETNA, with concentration on FCAPS. The second part focuses on evaluating ETNA implementation. Third part contrasts ETNA from an architectural standpoint to MPLS and other competing transport solutions. In the final section a few words is said about the showcase demonstrator.

5.1 ETNA Architecture Evaluation

During literature review we saw that network management has many established approaches. The FCAPS model concentrates purely on managing technical contraptions. The TMN model can be used to give arbitrary network performance indicators some meaning in business process context. The ITIL model can be used to ensure controlled transport service delivery and management throughout service lifespan.

Applying TMN or ITIL into ETNA scenarios is harder because this would require explicit knowledge about internal processes of a specific operator in order to align the necessary elements from each model. Thus we do not address such items here. However, we can apply FCAPS to ETNA framework by distinguishing key elements related to the five FCAPS areas, and as a result get some measurable items. This is shown in Table 5.1. As for faults, ETNA architecture has defined OAM probes which are generated on FE level. If probe generation interval is set sufficiently low, we are able to detect a missing probe on receiving side of the tunnel and alert the sending end to switch to backup route in sub-50ms time, equivalent to recovery capabilities of SDH. This means that

$$OAM_{int} + OAM_{det} + OAM_{gen} + OAM_{prop} + OAM_{proc} + OAM_{ps} < 50ms \quad (5.1)$$

Table 5.1: ETNA FCAPS.

Name	**Function**
Fault	OAM probing on FE level ⇒ theoretical sub-50ms recovery Control plane MS DB, path computation & signalling support protection routes
Configuration	Domain configuration in MS Network DB NE configuration in NE-local DB Configuration data partially as static MS SOAP Interface attributes, partially in XML descriptions
Accounting	Client information in MS DB Mechanisms for gathering real-time traffic information & usage statistics not implemented
Performance	Mechanisms for gathering network performance information not implemented Control plane can be integrated with 3rd party performance measurement systems
Security	Not addressed in ETNA

Where OAM_{int} is probe interval, OAM_{det} missing probe detection time, OAM_{gen} alert generation time, OAM_{prop} alert propagation time, OAM_{proc} alert procesing time and OAM_{ps} is protection switching time for backup tunnel. However, this does not hold if the OAM alert is sent to the MS through the same links where the possible fault has occurred; in this case we must either simply wait for both ends to notice the missing OAM or design the tunnels so that management alerts are always sent via paths that are link-disjoint from actual traffic paths. Currently the management system does not support this sort of reverse alert tunnel resiliency.

Regarding configuration we have everything domain-related configured in MS DB and tunnel/forwarding entry -related in NE DB. Depending on DB implementation configuration can either be kept in memory, storage or both. Frame classifiers and mappers are stored in XML format and can thus be changed dynamically. Other configuration parameters are a static part of the WSDL interface and DB backend, which is a point of improvement.

The Network DB contains customer and CI profiles for Accounting purposes. Apart from that no traffic measuring, billing etc. functionality was implemented. For the same reason, performance measurements are not possible with current intra-domain architecture.

Security was dropped from ETNA work packages in quite early stages of the project, thus the intra-domain control plane does not address security issues. Of course, security is provided in the same manner as conventional VLANs produce it; a frame can progress through the network only if it has correct fields as defined by the Customer Instance.

Next we will evaluate the ETNA implementation.

5.2 ETNA Implementation Evaluation

There are many things about the ETNA intra-domain control plane implementation which can be considered successes. For one, we described with BGU the classifier, mapper and forwarder XMLs that could be inserted on-the-fly from control element to the generic forwarding element, thus giving the forwarding element capability and understanding of what types of frames to forward, how, where and based on what criteria. Besides support for various IEEE frame types, we defined the custom "ETNA core" frame formats for intra-domain traffic transport. This implementation shows that the ETNA framework is capable of transporting any kinds of frames, standard or nonstandard. Adding new frame transmission capability to the system running on GFEs is achieved by simply writing a new XML description for the frame type.

Our signalling implementation also allows for provisioning and signalling both E-lines and E-trees from the control plane. For a protected E-line this yields unidirectional primary and protection paths to both directions. For E-trees this means signalling a tree first from source to one destination as primary path and then subpaths with respect to primary path and other destinations. The Path Computation Element was capable of calculating rudimentary protection paths also for multipoint configurations, but as these were not guaranteed to be link-disjoint we do not consider this a merit. The PCE supports dynamically changing the path computation algorithm, which was tested with two separate algorithms with zero disruption in service. Lastly we succeeded in defining both the northbound interface for instructing the control plane, and the southbound interfaces required for the control plane to operate with the data plane. Now any of the three components can, in theory, be replaced with another system that fullfills the specifications of the interface.

Much room for improvement still exists. For one, we should add more support for forwarding plane capability learning in the management system and control plane in general. Because the forwarding plane implementation did not support ForCES queries, we did not implement a method for propagating forwarding element information from control elements to the management system. Only the relevant columns in the management system database were done, should this information some day be transported all the way upstream. In addition, the classifier, mapper and forwarder XMLs are not currently propagated from the management system to the control elements, but instead exist preconfigured on CE-local storage.

Many other things could be also improved and finished in the management system. Currently most lacking is Control Element Manager, the listener for events signalled from the control elements. Completing it is a necessity for any OAM triggers to be recognized and recorded in the management system. Likewise, the current topology information collector implementation does not react to all changes in the network topology, and so has room for improvement. The whole topology collection process would benefit if we changed the Quagga-based -solution to a more lightweight one. Furthermore, the current management system database does not associate the used path computation algorithm to tunnels, but instead enforces the single algorithm selected at any given time as a one-size-fits-all solution to all computations. What this means is that if a path should ever need to be recalculated, a different algorithm might be used than what was originally requested for the tunnel.

The SOAP interface would most likely benefit if we would truncate various now-mandatory parameters from certain functions, and instead allow transfering same kind of XML descriptions as we are now currently doing with frame-related methods. More importantly, the object-oriented nature of SOAP is not currently utilized to its full potential, which is another improvement area. Unfortunately the rewrite required by these improvements would mean temporarily breaking the functionality of the existing user interfaces until they are updated.

Currently the PCE result wrapper classes contain Explicit and Secondary Explicit Route Objects (ERO, SERO) which describe the routes returned from path computation. When signalling the tunnel these objects can be inserted into mRSVP messages as is. This can be considered a bad thing from a software engineering standpoint as it ties the PCE implementation into one particular signalling protocol but a good thing from performance standpoint as it makes the overall implementation faster.

Another possible improvement would be an authentication module inside the Management Logic, as currently the management plane request are processed to with blind trust.

5.3 CET Candidates Versus ETNA

MPLS is the current incumbent in packet-based transport services. Therefore we begin our evaluation of ETNA versus CET candidate technologies by contrasting MPLS to ETNA.

The biggest difference is that MPLS has an IP-based control plane, whereas ETNA control plane architecture is agnostic to network layer. Proponents of MPLS say it *is* possible to traffic engineer Label Switched Paths “behind” the distributed IP routing process of MPLS. However, this does not scale to large provider networks without a centralized traffic engineering system. The motivation for vendors to develop such a system has been low because the actual problems lie within the

MPLS architecture itself. Paths beyond the minimum shortest path are interesting only if protection needs to be warranted. Protection is not possible without OAM monitoring capabilities, which in turn are difficult to realize in MPLS due to LSPs being disjoint from the IP control plane. In contrast, ETNA architecture allows control signalling to traverse either the same path or different path as the actual OAM traffic. In addition, generating OAM probes and switching traffic to protection paths is possible within a timeframe equivalent to SDH protection. These backup paths are calculated and pre-signalled by the control plane Management System for each service which requires protection. Another thing the MPLS architecture does not offer is domain wide capability and capacity discovery. Provisioning large scale transport services requires detailed information of the capacity reservations made for constant and variable bit rate services throughout the domain, and the ability to adjust traffic according to this information. Usage of LDP or RSVP-TE in an MPLS network is not up to par with this requirement. Information about available bandwidth and interface capabilities in general needs to flow all the way from Network Element to the Management System. This is builtin to the ETNA architecture.

ELS, as opposed to MPLS, is a transport oriented packet switched technology by nature. While it is an attractive thought to be able to offer 16 million service instances, ELS does not offer a centralized NMS for assigning and signalling these Ethernet switched "tunnels" inside an ELS domain. Therefore the scalability in ELS is currently lackluster. Problems also exist in ELS multipoint forwarding. It can be said that ELS and ETNA are not even competitors but could support each other; it is only a matter of integration work to use an ETNA-based control plane to deliver Carrier Ethernet Transport services on top of a data plane forwarding frames with ELS.

PBB-TE offers, to a large extent, same kind of forwarding functionality as ETNA. Both ETNA and PBB-TE are able to support multiple forwarding results for a given frame, which simplifies forwarding because the same engine can be used for both point-to-point and point-to-multipoint services. Much like ETNA, PBB-TE aims to provide QoS-constrained transport services which are traffic engineered end-to-end. The central point here is that PBB-TE lacks a control plane architecture altogether whereas ETNA provides a robust control plane for provisioning transport services. Additionally, for example interworking of multicast IGMP and PBB-TE has not yet been verified. ETNA supports IGMP on Network Service Layer, but much of the functionality required for this has not yet been implemented.

MPLS-TP is probably the toughest competitor to ETNA. It offers the capability to transport any kind of L1-L3 service. L1 traffic requiring synchronization is carried on top of MPLS pseudowires with various synchronization mechanisms currently proposed. ETNA control or data plane architectures do not have strongly identifiable modules responsible for synchronization. If the ETNA data plane is extended to run synchronized L1 traffic, it can do so, but the control plane merely tells between which interfaces and through which nodes the bitstream will flow, not for example where the reference clocks are provided and what is the reference clock hi-

erarchy. Otherwise the properties of MPLS are akin to ETNA regarding multipoint support and transport properties. A comparison of the evaluated CET candidates can be seen in Table 5.2.

Table 5.2: Comparison of CET candidate technologies, adapted from [3].

	IP / MPLS	**ELS**	**PBB-TE**	**MPLS-TP**	**ETNA**
Transport oriented	No	Yes	Yes	Yes	Yes
Scalable	To some extent	To some extent	Yes	Yes	Not verified
Multipoint support	Yes	To some extent	To some extent	Yes	Yes
Standardized or in the process	Yes	Expired draft	Yes	Yes	Not yet

5.4 ETNA Showcase

The focal point of the ETNA project was the demonstrator event or showcase which was held in Ipswich, UK on 25th November 2009[1]. In the showcase, various ETNA intra- and inter-domain networking scenarios were demonstrated. Intra-domain functionality utilized our control plane and showcased control plane properties from multipoint support to signalling and protection capability. Inter-domain scenarios consisted of several interjoined ETNA domains which were provisioning services spanning multiple such domains. This was achieved with Ethos' centralized inter-domain provisioning system, which relays subscriptions to BGU's inter-domain OSSs, which uses the SOAP interface to request tunnels from ComNet's intra-domain control plane Management Systems, which in turn calculate and signal tunnels to CEs. Finally, the CEs install corresponding forwarding entries into BGU GFEs. Details about ETNA demo arrangements can be found from ETNA documentation. [8]

Now that ETNA evaluation is finished, it is suitable to give some conclusions and directions for future research for this work. This is done in the next chapter.

[1]Videos about the showcase are available in `http://vimeo.com/showcase`

Chapter 6

Conclusions and summary

In this chapter, conclusions of the research related to ETNA intra-domain control plane management and a general summary are given.

6.1 Conclusions

Currently there are many legacy systems being used stacked on top of each other to deliver Carrier Ethernet services. This yields higher cost per bit than a unifield solution would. Ethernet is getting pressure from all directions to solve this problem by morphing into a transport technology.

Two years ago we at Comnet started building a centralized control plane for a carrier grade Ethernet transport network. Two years later we have accomplished several things. We implemented Control Element software which is capable of signalling multipoint and protected point-to-point paths. Additionally, together with BGU we defined various frame type XML descriptions which describe what sort of customer and OAM frame types we support and how they are forwarded. We also implemented a ForCES connector which can proxy ForCES messages between the a Control Element and a Forwarding Element. The TUN/TAP[1] driver was used to transfer Ethernet interfaces of a network processor via ForCES to a Control Element which resides on a separate machine—this way the CE could use the FE's interfaces as its own. And lastly, we designed and implemented a centralized, web service -based management system which is able to do book-keeping about domain topology and capacity reservations. Based on this book-keeping and selected quality constraints, the control plane Path Computation Element can calculate paths for tunnels inside the domain.

During this time we also learned a lot. We learned that a network processor is quite a versatile forwarding platform when programmed with skill, and can be a valuable aid in developing proof-of-concept networks. We learned that software integration is a tedious task during which unexpected bugs or design errors may

[1] http://vtun.sourceforge.net/

arise no matter how well the integration is planned beforehand. We also learned that open source is not a free lunch; even with various free libraries and tools, certain overhead is consumed in first learning the software and then putting it to use. The best lesson I personally gained from ETNA is that network technologies are much more elastic than they seem. Under a market which is constantly looking for ways to cut costs and optimize profit, a technology must be able to re-invent itself if users are starting to question its profitability. ATM failed to accomplish this. IPv4—something which I personally, not long ago, believed will exist forever—will get a breather to its address space exhaustion problem with IPv6 but time will tell if this is only artificial respiration. MPLS working groups are pushing hard to convert MPLS into a full-blown carrier transport. And during all this, Ethernet is being actively developed into a transport network technology.

There are various apporaches, best practices and standards related to network management. They should be utilized and combined throughout the network design and construction process. Standards emphasise that systems should be interoperable and able to encapsulate and convey information to higher-order systems. When information is sent upstream from network element level and filtered accordingly, it is easier to predict and evaluate the business impact a disruption in network services might have.

It seems that the gap from local area Ethernet to Carrier Ethernet is starting to close. It is soon up to operators to decide which technology will eventually give them best cost per bit - the "plug-and-play" solution of hierarchical provider bridged networks, the familiar IP/MPLS-driven solution for which many carriers already have existing equipment in place, or a traffic engineered, centralized transport network such as MPLS-TP, PBB-TE or ETNA. At this point the only certain thing seems to be that running IP/MPLS will come to its end, because a control plane based on a technology with a death warrant due to legacy features and address space exhaustion is not going to sustain.

6.2 Further Research

There are quite many research subjects in continuing the work done in ETNA. Comnet could test the control plane implementation with other data planes to see if we could, for example, create normal MPLS services. In addition, inter-domain scenarios where topology is discovered dynamically should be researched. Systems requesting services from an ETNA domain may be capable of communicating with multiple such transport service domains. An example use case for this would be an auctioning system for inter-domain transport services. The auctioning system would query Management Systems of individual domains about their edge interfaces' utilization and connectivity to adjacent domains. In this way the auctioning system could form an inter-domain transport network topology. However, a protocol for adjacent inter-domain CET discovery is required for this to work. Dynamically propagating frame type information from MS to CE with mRSVP is an additional

development subject.

Scalability is also an interesting research issue; ETNA test domains, be they physical or virtual, were built in very conservative sizes (1-100 nodes) which didn't present any problems for path computation. We should find the stress limits for our Path Computation Element and signalling, especially for multipoint services. PCE and the Network DB should also be extended to support multiple algorithms simultaneously. Additionally, once CE status reporting protocols have been implemented, it should be researched how much bandwidth the overall signalling, maintenance, and various other OAM operations consume.

6.3 Summary

In the beginning of this thesis we see that the current means of creating transport services for both packet and circuit traffic are becoming too complex and expensive for operators to run. Additionally we suggest that Ethernet is going to be the dominant transport technology when IP / MPLS is starting to come to the end of the line. After this we set out to study the management of intra-domain carrier grade Ethernet, focusing on control plane management, with the intention to build a management system that can be used to deliver these Carrier Ethernet services in some form.

Next we introduce the globally most dominant forms of Ethernet as LAN technology, which builds the motivation and demand for Carrier Ethernet services. Thereafter we continue to give an overview how these services have been standardized and also how they are being delivered with non-Ethernet equipment. In the latter part of the chapter we introduce Carrier Ethernet Transport, an all-encompassing Ethernet-based transport solution for carrying any type of traffic. Additionally, some requirements for CET service assurance are covered. We conclude the chapter by introducing the most dominant CET candidates existing today.

In Chapter three we discuss how network management relates to the carrier business infrastructure. Additionally we study how existing network management standards help in aligning network management processes to this infrastructure. Based partially on the standards and other literature we review the candidates for the northbound interface of our management system.

In the fourth chapter we discuss aspects of ETNA architecture. We introduce the transport, transport service and network service layer concepts in ETNA. Thereafter an overview about the intra-domain control plane management system is given. First the management system architecture is described. After this the implementation specifics and interfaces are discussed. In the final section some analysis about the ETNA architecture is done.

In the final chapter we find that as an overall the management system filled its purpose. It is capable of provisioning point-to-point and multipoint transport services, and interact with the Management Plane, Control Elements and Forward-

ing Elements to establish a Carrier Ethernet Transport service. Additionally it is discussed what future research could be warranted on the progress made so far.

Bibliography

[1] Dieter Beller and Rolf Sperber. 2009. *MPLS-TP – The New Technology for Packet Transport Networks.* 2. DFN Forum, München.

[2] R. Bhandari. 1999. *Survivable Networks: Algorithms for Diverse Routing.* Kluwer Academic Publishers.

[3] Achim Autenrieth Claus G. Gruber. 2008. *Carrier Ethernet Transport in Metro and Core Networks, tutorial.*

[4] ETNA Consortium. 2008. *Ethernet Transport Networks, Architectures of Networking. Work Package 1 Deliverable 1.1. Requirements, specification and analysis.* URL `http://www.ict-etna.eu/documents/ETNAWP1FinalD1.1.pdf`. Referenced 3.3.2010.

[5] ETNA Consortium. 2008. *Ethernet Transport Networks, Architectures of Networking. Work Package 2 Deliverable 2.1. Network Architecture.* URL `http://www.ict-etna.eu/documents/ETNAWP2NetworkandServiceArchitecture-D2.1R2-Issue2.pdf`. Referenced 3.3.2010.

[6] ETNA Consortium. 2008. *Ethernet Transport Networks, Architectures of Networking. Work Package 4 Deliverable 4.1. Requirements, Implementation, Architecture, and Functionality.* URL `http://www.ict-etna.eu/documents/ETNAReportoftherequirements, implementationarchitectureandmodels-D4.1-R1.pdf`. Referenced 28.3.2010.

[7] ETNA Consortium. 2008. *Ethernet Transport Networks, Architectures of Networking. Work Package 4 Deliverable 4.2. Report on the implementation details.* URL `http://www.ict-etna.eu/documents/ETNAD4.2Reportontheimplementationdetails.pdf`. Referenced 28.3.2010.

[8] ETNA Consortium. 2010. *Ethernet Transport Networks, Architectures of Networking. Work Package 6 Deliverable 6.1. Showcase Report.* URL `http://www.ict-etna.eu/documents/ETNA%20D6.1%20Showcase%20Report%20V1.4.pdf`. Referenced 3.3.2010.

[9] F. Le Faucheur, L. Wu, B. Davie, S. Davari, P. Vaananen, R. Krishnan, P. Cheval, and J. Heinanen. 2002. Multi-Protocol Label Switching (MPLS) Support of Differentiated Services. RFC 3270 (Proposed Standard). URL `http://www.ietf.org/rfc/rfc3270.txt`.

[10] Don Fedyk and David Allan. 2008. *Ethernet Data Plane Evolution for Provider Networks*. IEEE Communications Magazine.

[11] Norman Finn. 2006. MAC address transparency. URL `http://www.ieee802.org/1/files/public/docs2006/ad-nfinn-mac-address-transparency-0506-v2.pdf`. IEEE 802.1 Interim, Beijing. Referenced 3.3.2010.

[12] Ran Giladi. 2008. *Network Processors*. Morgan Kaufmann.

[13] Ran Giladi and Niv Yemini. 2009. *A Programmable, Generic Forwarding Element Approach for Dynamic Network Functionality*. Ben Gurion University.

[14] Jan Gröndahl. 2010. *Implementation and Evaluation of a Network Element Control Protocol*. Master's thesis, Aalto University School of Science and Technology.

[15] Timo-Pekka Heikkinen. 2008. *Testing the Performance of a Commercial Active Network Measurement Platform*. Master's thesis, TKK.

[16] J. Won-Ki Hong. 2003. *Telecommunications Management Network*. Lecture slide.

[17] IEEE. 2006. *Standard for local and metropolitan area networks, virtual bridged local area networks*. IEEE Std 802.1Q-2005 .

[18] IEEE. 2008. *Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems* 1588v2.

[19] Fujitsu Network Communications Inc. 2006. Ethernet Service OAM: Overview, Applications, Deployment, and Issues. White Paper.

[20] Network Instruments. 2008. *State of the Network Global Study*.

[21] ITU-T. 1993. *Recommendation M.3010: Principles for a Telecommunications management network*.

[22] ITU-T. 1993. *Recommendation M.3400: TMN Management functions*.

[23] ITU-T. 2007. *Recommendation M.3050: Enhanced Telecom Operations Map*.

[24] A. Kirstädter, C. Grubera, J. Riedla, and T. Bauschert. 2006. *Carrier-Grade Ethernet for Packet Core Networks*.

[25] Olli-Pekka Lamminen, Marko Luoma, Jukka Nousiainen, and Taneli Taira. 2009. *Control Plane for Carrier-Grade Ethernet Network*.

[26] Juha-Matti Lehtonen et al. 2004. *Tuotantotalous*. WSOY.

[27] Mo Li and Kumbesan Sandrasegaran. 2005. *Network Management Challenges for Next Generation Networks*. Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary.

[28] Aref Meddeb. 2005. *Why Ethernet WAN Transport?* IEEE Communications Magazine.

[29] Robert M. Metcalfe and David R. Boggs. 1976. *Ethernet: distributed packet switching for local computer networks*. Commun. ACM 19, no. 7, pages 395–404.

[30] Kundan Misra. 2004. *OSS for Telecom Neworks. An Introduction to Network Management.* Springer.

[31] Cisco Networks. 2009. *Understanding MPLS-TP and Its Benefits*. White Paper.

[32] Mark Norris. 2003. *Gigabit Ethernet - Technology and applications*. Artech House.

[33] Office of Government Commerce. 2007. *Official Introduction to the ITIL Service Lifecycle*. Stationery Office Books.

[34] Finlex / Edita Oyj. 2006. Osakeyhtiölaki 21.7.2006/624.

[35] Jeff Parker. 2005. *FCAPS, TMN & ITIL - Three Key Ingredients to Effective IT Management. Enterprise Management System White Paper*. page 7.

[36] Nurit Sprecher Philippe Klein. 2006. *Provider Ethernet VLAN Cross Connect*.

[37] Samer Salam and Ali Sajassi. 2008. *Provider Backbone Bridging and MPLS: Complementary Technologies for Next-Generation Carrier Ethernet Transport*. IEEE Communications Magazine.

[38] Oscar Santolalla. 2009. *Implementation of IS-IS Extensions for Routed End-to-end Ethernet*. Master's thesis, Helsinki University of Technology.

[39] David Stokes and Yoav Cohen. 2009. *MEF Overview and Carrier Ethernet Introduction*.

[40] Cisco Systems. 2005. Cisco IOS Software Releases 12.0 - Stacked VLAN Processing. URL `http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/qinq.html`. Referenced 3.3.2010.

[41] Taneli Taira. 2010. *A Signaling System for Ethernet Transport*. Master's thesis, Aalto University School of Science and Technology.

[42] Tuomas Toropainen. 2008. *A Routing Protocol for Ethernet Transport*. Master's thesis, Helsinki University of Technology.

[43] Kenneth J. Turner, Evan H. Magill, and David J. Marples. 2004. *Service Provision*. John Wiley & Sons, Ltd.

[44] Reza Vaez-Ghaemi. 2009. *Ethernet OAM test applications*. Lightwave. URL `http://lw.pennnet.com/display_article/353044/13/ARTCL/none/none/1/Ethernet-OAM-test-applications/`. Referenced 3.3.2010.

[45] L. Yang, R. Dantu, T. Anderson, and R. Gopal. 2004. Forwarding and Control Element Separation (ForCES) Framework. RFC 3746 (Informational). URL `http://www.ietf.org/rfc/rfc3746.txt`.

Appendix A

Building and Running Instructions

In this Appendix some of the commands required to build and run the network management software that was implemented in ETNA are described. A view on how the source code for this software is organized in the Subversion repository can be seen in Listing 1.

Listing 1 Structure of the management system code base.

```
.
|-- ce
|-- isis
|-- ms
|   |-- branches
|   |-- db
|   |-- pce
|   |-- python-gui
|   |-- soap_handler // (management logic)
|   |-- sql_api
|   |-- topology_export
|   |-- web_if // (web UI)
|   `-- wsdl
`-- utils
```

The `ce`, `isis`, and `utils` repositories contain code related to control element functionality, topology discovery and miscellaneous utilities, respectively. The `ms` repository contains dedicated directories for each part of the management system and relevant interfaces. Some of the project inside the `ms` repository include headers and libraries from other repositories and projects. The easiest way to ensure proper building is to create symbolic links to `ce`, `utils` and `ms` in each project directory.

Management Logic

The management logic program can be invoked with `mlogic OPTIONS`. Without options the usage information shown in Listing 2 is provided.

Listing 2 Usage of Management Logic.

```
Usage: mlogic OPTIONS
OPTIONS contains one or more of the following:

        [-d]            invoke the program as a daemon
        [-p port]       TCP port of RSVP recipient
        [-o host]       SQL host (IP address)
        [-u user]       SQL username
        [-w password]   SQL password
        [-c database]   SQL database
        [-r port]       SQL port number
        [-f]            Set per-SOAP-request forking on
        [-g filename]   Location of re2ee.conf
        [-s 0...3]      Interface mode:
                        1 - listen to external SOAP interface
                        2 - listen to local SOAP interface
                        3 - listen to both SOAP interfaces

Please note that with -d you must use -s1/2/3.
```

The service description in `wsdl` directory, `sql_api` library, and various libraries in `ce` and `utils` directories must be built before building Management Logic.

Database schema

The management system DB schema can be loaded into a MySQL database as shown in Listing 3. The schema itself can be found from ETNA Deliverable 4.2. It has been tested with MySQL 5.1 and should support any equivalent or newer version.

Listing 3 Loading command for Management System Database SQL schema.

```
$ mysql -u username -p < msdatabase.sql
```

Topology Export

The usage for topology export program, which exports the Topology Discovery Link State Database into MS Network DB, is shown in Listing 4.

Listing 4 Usage of Topology Export.

```
Usage: topology_export OPTIONS
OPTIONS containing:

        [-f filename]   location of IS-IS LSDB file, which MUST
                        contain the RE2EE extensions defined and
                        implemented at Aalto University.
```

Topology export depends on `sql_api`.

Topology Discovery

Starting the topology discovery is a two part process where first the bulk routing message forwarding engine (zebra) is initialized, followed by the specific routing protocol daemon, in this case isisd. One option to do this is to use a bash script as provided in Listing 5. The daemon also presumes that a third file called re2ee.conf is present in /usr/local/etc. This file contains node-local configuration options such as the initial NSAP address of the node.

Listing 5 Launching of Topology Discovery.

```
#!/bin/bash
zebra/zebra -d -u root -f /usr/local/etc/zebra.conf
sleep 1;
isisd/isisd -d -u root -f /usr/local/etc/isisd.conf
```

Topology Discovery does not have any build dependencies to other control plane modules.

Building

All software implemented by the author is built by invoking `make` in the subdirectory of each project, given that GNU make, GNU Compiler Collection (or at least g++) and required libraries are installed. Information about libraries needed can be read in each projects' Makefile.

Appendix B

Screenshots

This Appendix contains screenshots of ETNA control plane management system UI implementations.

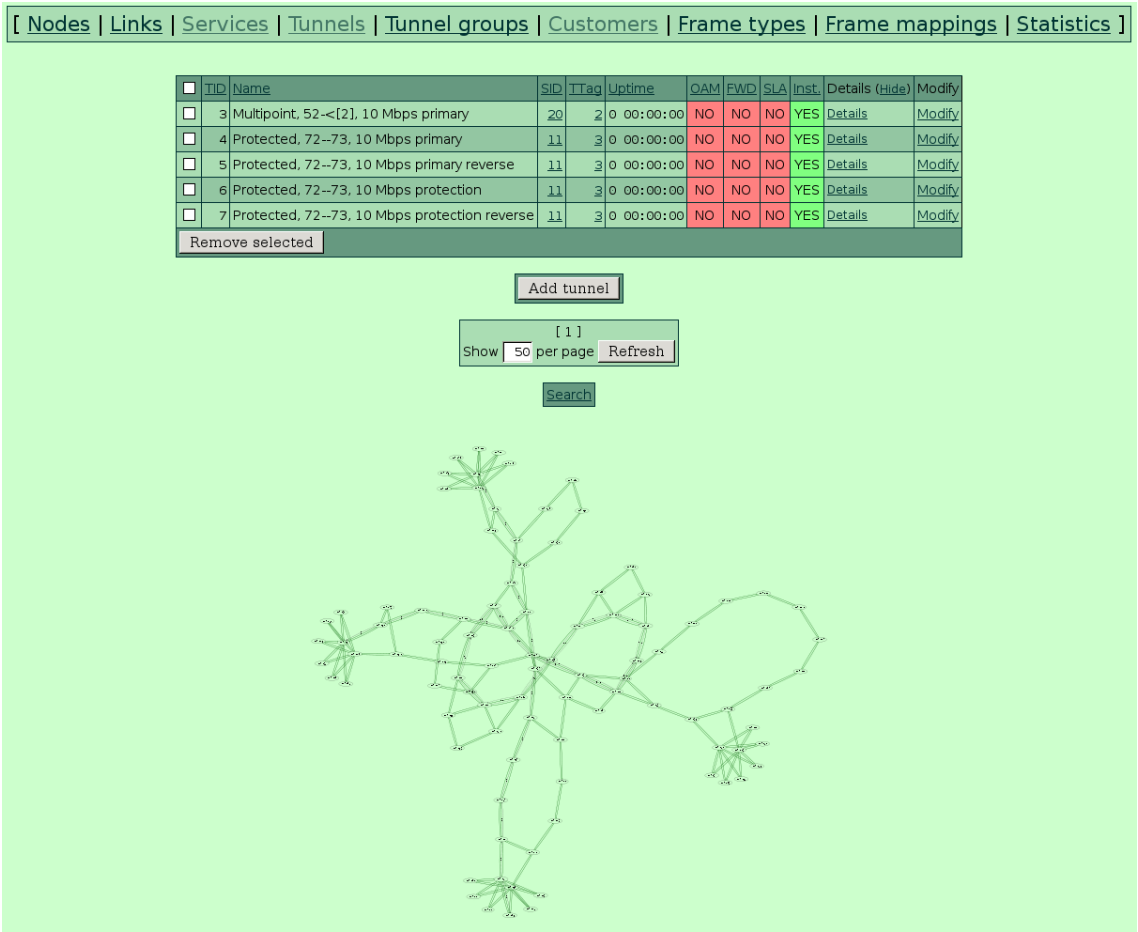


Figure B.1: Web User Interface of ETNA Control Plane Management System.

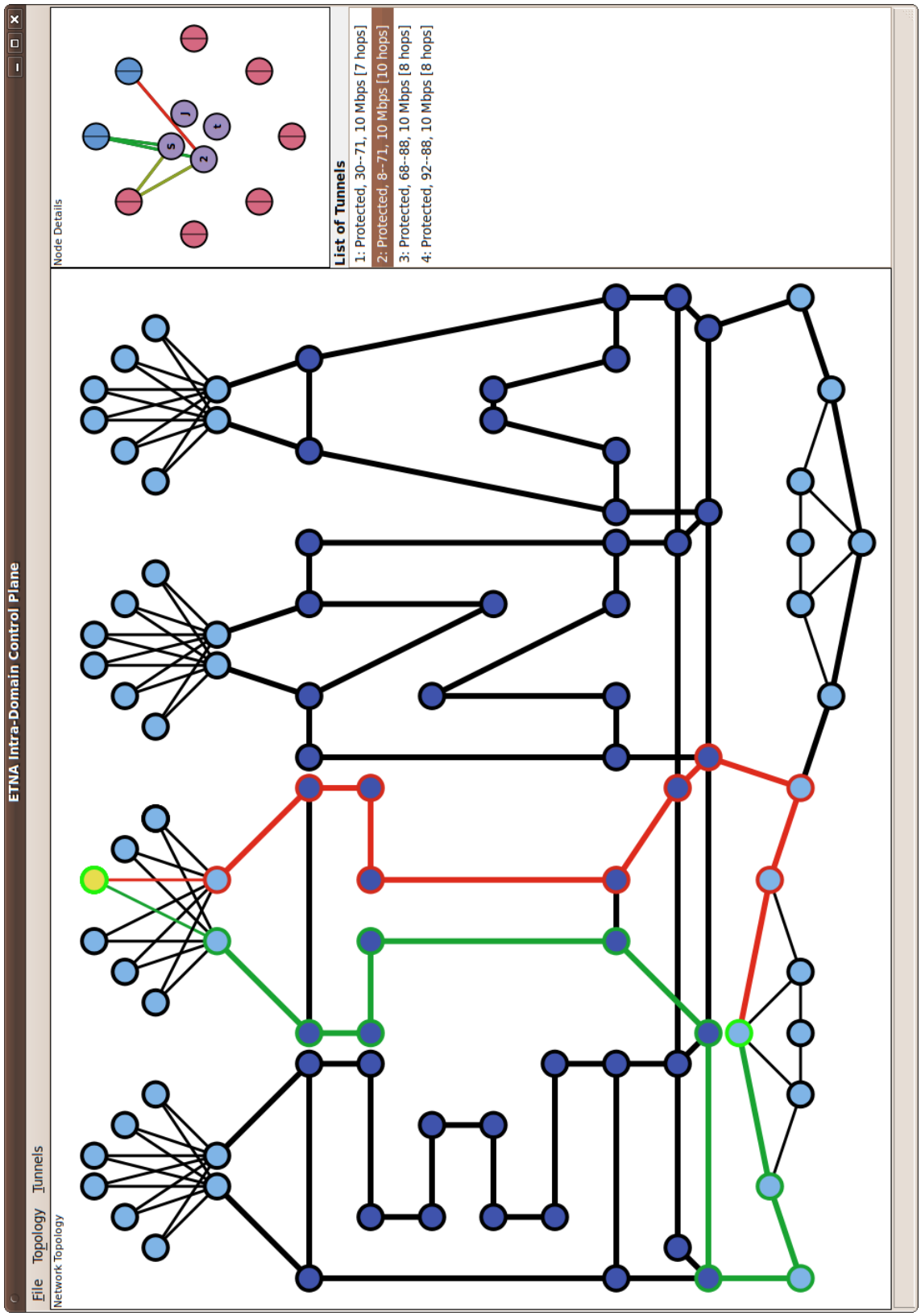


Figure B.2: Python User Interface of ETNA Control Plane Management System.

Appendix C

Signalling Diagrams

This Appendix contains signalling diagrams of ETNA management scenarios.

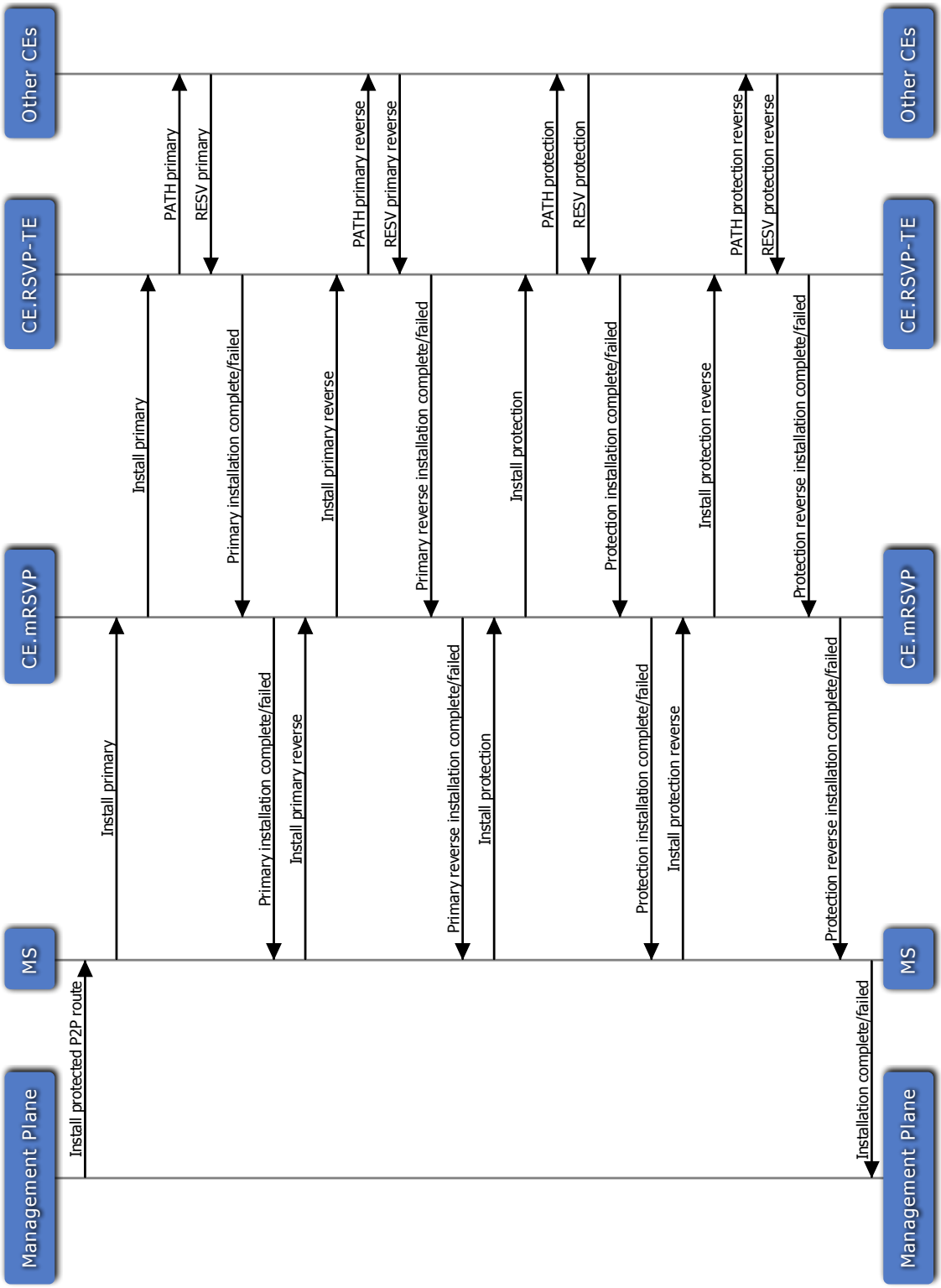


Figure C.1: Signalling a protected point-to-point tunnel with mRSVP and RSVP-TEEth.

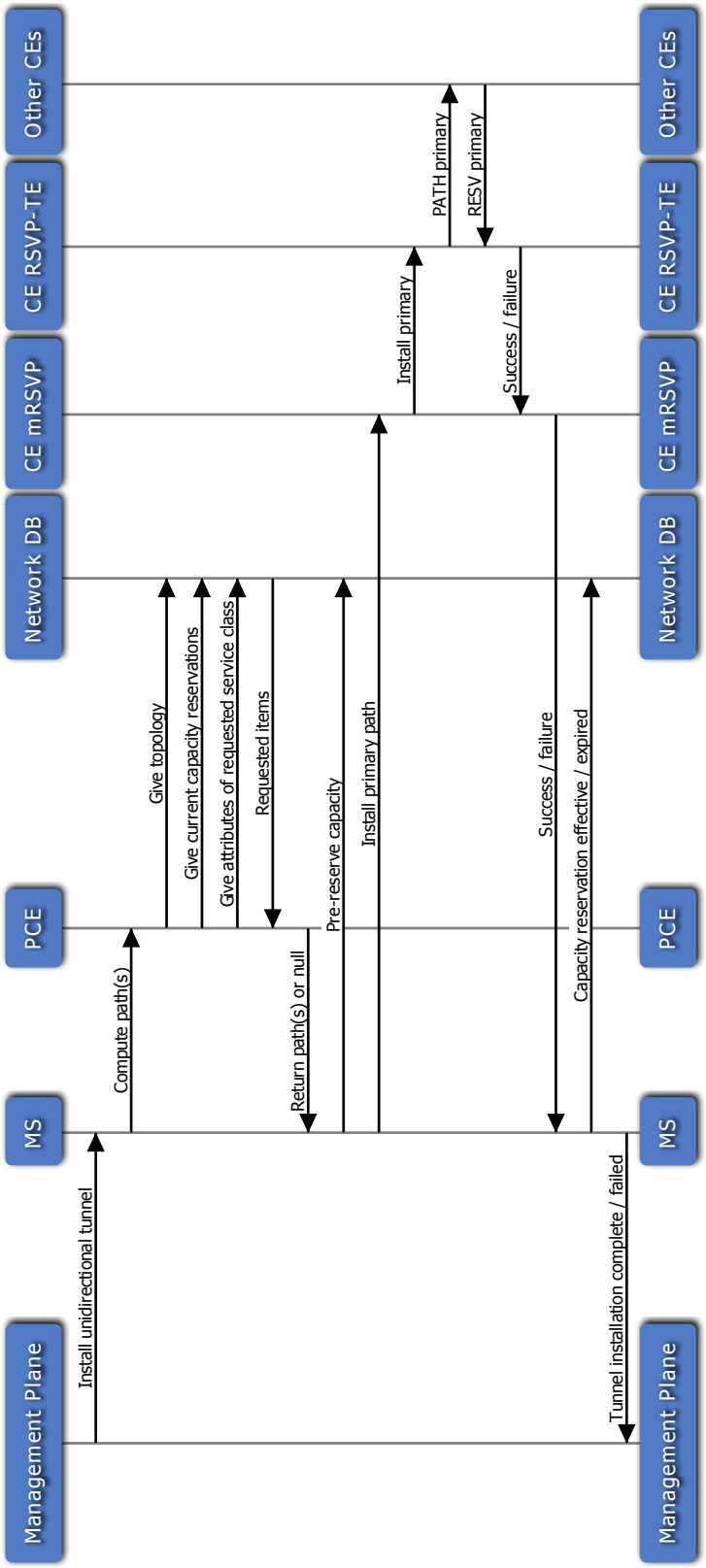


Figure C.2: Signalling an unprotected half-duplex point-to-point tunnel with mRSVP and RSVP-TEEth.