Ari Lappeteläinen

Equal Cost Multipath Routing in IP Networks

Faculty of Electronics, Communications and Automation

Thesis submitted for examination for the degree of Master of Science in Technology.

17.3.2011

Thesis supervisor:

Prof. Raimo Kantola

Thesis instructor:

M.Sc.(Tech.) Pasi Kinnari

Aalto University
School of Science
and Technology

AALTO UNIVERSITY
SCHOOL OF SCIENCE AND TECHNOLOGY

ABSTRACT OF THE
MASTER'S THESIS

Author: Ari Lappeteläinen

Title: Equal Cost Multipath Routing in IP Networks

Date: 17.3.2011 Language: English Number of pages:12+85

Faculty of Electronics, Communications and Automation

Professorship: Department of Communications and Networking Code: S-38

Supervisor: Prof. Raimo Kantola

Instructor: M.Sc.(Tech.) Pasi Kinnari

Increasing efficiency and quality demands of services from IP network service providers and end users drive developers to offer more and more sophisticated traffic engineering methods for network optimization and control. Intermediate System to Intermediate System and Open Shortest Path First are the standard routing solutions for intra-domain networks. An easy upgrade utilizes Equal Cost Multipath (ECMP) that is one of the most general solutions for IP traffic engineering to increase load balancing and fast protection performance of single path interior gateway protocols.

This thesis was written during the implementation process of the ECMP feature of Tellabs 8600 series routers. The most important parts in adoption of ECMP are changes to shortest path first algorithm and routing table modification in the control plane and implementation of load balancing algorithm to the forwarding plane of router.

The results of the thesis and existing literature prove, that the load balancing algorithm has the largest affect on traffic distribution of equal cost paths and the selection of the correct algorithm is crucial. Hash-based algorithms, that keep the traffic flows in the same path, are the dominating solutions currently. They provide simple implementation and moderate performance. Traffic is distributed evenly, when the number of flows is large enough.

ECMP provides a simple solution that is easy to configure and maintain. It outperforms single path solutions and competes with more complex MPLS solutions. The only thing to take care of is the adjustment of link weights of the network in order to create enough load balancing paths.

Keywords: ECMP, Load balancing, Traffic engineering, Routing

AALTO-YLIOPISTO
TEKNILLINEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Ari Lappeteläinen

Työn nimi: Monipolkureititys IP verkoissa

Päivämäärä: 17.3.2011 Kieli: Englanti Sivumäärä:12+85

Elektroniikan, tietoliikenteen ja automaation tiedekunta

Professuuri: Tietoverkkotekniikka Koodi: S-38

Valvoja: Prof. Raimo Kantola

Ohjaaja: DI Pasi Kinnari

IP verkkojen palveluntarjoajat ja loppukäyttäjät vaativat yhä tehokkaampia ja parempilaatuisia palveluita, mikä vaatii tuotekehittäjiä tarjoamaan hienostuneempia liikennesuunnittelumenetelmiä verkon optimointia ja hallintaa varten. IS-IS ja OSPF ovat standardiratkaisut hoitamaan reititystä pienissä ja keskisuurissa pakettiverkoissa. Monipolkureititys on melko helppo ja yleispätevä tapa parantaa kuorman balansointia ja nopeaa suojausta tällaisissa yhden polun reititykseen keskittyvissä verkoissa.

Tämä diplomityö kirjoitettiin aikana, jolloin monipolkureititys toteutettiin Tellabs-nimisen yrityksen 8600-sarjan reitittimiin. Tärkeimpiä kohtia monipolkureitityksen käyttöönotossa ovat lyhyimmän polun algoritmin muokkauseen ja reititystaulun toimintaan liittyvät muutokset ohjaustasolla sekä kuormanbalansointialgoritmin toteutus reitittimen edelleenkuljetustasolla.

Diplomityön tulokset sekä olemassaoleva kirjallisuus osoittavat, että kuormanbalansointialgoritmilla on suurin vaikutus yhtä hyvien polkujen liikenteen jakautumiseen ja että oikean algoritmin valinta on ratkaisevan tärkeää. Hajakoodaukseen perustuvat algoritmit, jotka pitävät suurimman osan liikennevuoista samalla polulla, ovat dominoivia ratkaisuja nykyisin. Tämän algoritmityypin etuna on helppo toteutettavuus ja kohtuullisen hyvä suorituskyky. Liikenne on jakautunut tasaisesti, kunhan liikennevuoiden lukumäärä on riittävän suuri.

Monipolkureititys tarjoaa yksinkertaisen ratkaisun, jota on helppo konfiguroida ja ylläpitää. Suorituskyky on parempi kuin yksipolkureititykseen perustuvat ratkaisut ja se haastaa monimutkaisemmat MPLS ratkaisut. Ainoa huolehdittava asia on linkkien painojen asettaminen sillä tavalla, että riittävästi kuormantasauspolkuja syntyy.

Avainsanat: ECMP, Kuorman balansointi, Liikennesuunnittelu, Reititys

Preface

This work was done during the ECMP implementation project of Tellabs routers. I thank all people, who took part of this project and hence made this thesis possible. Ville Hallivuori gave lot of insight to this project and he deserves special thanks. I would like to thank all colleagues, who gave me practical guidance. I also thank Pasi Kinnari, who helped much on work schedules, and William Martin for proofreading the final version of the thesis manuscript.

Otaniemi, 2.3.2011

Ari A.L. Lappeteläinen

Contents

Acronyms

ABR	Area Border Router
ACK	Acknowledgement
ASBR	Autonomous System Boundary Router
ANSI	American National Standards Institute
ARP	Address Resolution protocol
AS	Autonomous System
ASIC	Application Specific Integrated Circuit
BGP	Border Gateway Protocol
CLI	Command Line Interface
CLNP	ConnectionLess Network Protocol
CLNS	ConnectionLess-mode Network Service
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSNP	Complete SNP
CSPF	Constraint-based Shortest Path First.
DA	Destination Address
DHTC	Dual hash table and counters algorithm
DIS	Designated IS (in IS-IS)
DR	Designated Router (in OSPF)
DV protocol	Distance Vector protocol
ECMP	Equal-Cost Multipath
EIGRP	Enhanced Interior Gateway Routing Protocol
ES	End System
FLARE	Flowlet Aware Routing Engine
FS	Fast Switching
FEC	Forwarding Equivalence Class
HRW	Highest Random Weight
ICMP	Internet Control Message Protocol
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IGRP	Interior Gateway Routing Protocol
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
IS	Intermediate System
IS-IS	Intermediate System-to-Intermediate System
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU-T	International Telecommunication Union, Telecommunication Standardization Sector
LAG	Link Aggregation Group
LACP	Link Aggregation Control Protocol

LAN	Local Area Network
LCC	LRU-based Caching with Counting
LCER	Link-Criticality-based ECMP routing
LDP	Label Distribution Protocol
LER	Label Edge Router
LSA	Link-State Advertisement
LSP (MPLS)	Label-Switched Path
LSP (IS-IS)	Link-state PDU
LSR	Label Switch Router
MAC	Medium Access Control
MPLS	Multiprotocol Label Switching
MPTCP	Multipath TCP
MTU	Maximum Transmission Unit
NCR	Non-congestion robustness (TCP)
NET	Network Entity Title
NLPID	Network Layer Protocol ID
NLSP	Netware Link Services Protocol
NPDU	Network Layer PDU
NPU	Network Processor Unit
NSSA	Not-so-stubby-area
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OSPF-OMP	OSPF optimized multipath
PDU	Protocol Data Unit
PEFT	Penalizing Exponential Flow-splitTing
PLR	Point of Local Repair.
PSNP	Partial SNP
PW	Pseudo Wire
PWE3	Pseudo Wire Emulation Edge to Edge
QoS	Quality of Service
RED	Random Early Detection
RFC	Request For Comments
RIB	Route Information Base
RIP	Routing Information Protocol
RSVP	Resource ReSerVation Protocol
RSVP-TE	Resource Reservation Protocol - Traffic Engineering
SA	Source address
SNP	Sequence Numbers PDU
SNPA	SubNetwork Point of Attachment
SPF	Shortest Path First
SPT	Shortest Path Tree
TCP	Transmission Control Protocol
TE	Traffic Engineering
TH	Table-based Hashing
THR	Table-based Hashing with Reassignments

TLV	Type, Length, Value
TTL	Time To Live
UDP	User Datagram Protocol
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol.
WAN	Wide Area Network

List of Figures

List of Tables

1 Introduction

The importance of IP networks has constantly increased during the past decade. Not only the size of the Internet including the backbone, access networks and number of connected devices has increased but also the requirements for network performance from end-users and network efficiency from Internet service providers (ISPs) and mobile backhaul service providers have increased.

Traffic engineering tries to optimize both network efficiency and the performance to the current network conditions. One traffic engineering method is Equal Cost Multipath (ECMP) that enables the usage of multiple equal cost paths from the source node to the destination node in the network. The advantage is that the traffic can be split more evenly to the whole network avoiding congestion and increasing bandwidth. ECMP is also a protection method, because during link failure, traffic flow can be transferred quickly to another equal cost path without severe loss of traffic.

Link-state routing protocols are nowadays the most commonly used way to find the shortest path from source to destination node in small and medium sized packet networks. These protocols, such as Open Shortest Path First (OSPF) or Intermediate System-Intermediate System (IS-IS), are capable of creating and maintaining several equal cost paths in their routing tables. The relatively small addition to the basic Dijkstra's algorithm is enough to adopt the ECMP feature.

The purpose of this thesis is to discuss the theoretical and practical issues of ECMP. The thesis was written during the ECMP implementation project of the Tellabs 8600 series routers. The author was responsible for higher abstraction layer parts of the software, that is related to logical ECMP group management. Important topics of the thesis are ECMP implementation into a real routing system and its relation to other protection and load balancing methods. The load balancing algorithm has the key role of achieving fast protection and well distributed load balancing. It is also relevant to discover the benefits of ECMP by comparison with traditional single path (SP) and MPLS-based routing. The study is concentrated on the usage of the OSPF and IS-IS protocols but also ECMP implementation for static routing is discussed.

In Chapter 2 the basics of IP routing are described. Chapter 3 introduces link-state protocols and the changes needed in those to adopt ECMP. Chapter 4 introduces different traffic engineering methods to distribute traffic more evenly in the network and to protect the network against failures. Chapter 5 describes the most relevant parts relating to ECMP implementation. Chapter 6 discusses the configuration and testing issues, Chapter 7 introduces the results and analysis and Chapter 8 presents the conclusion of the thesis.

2 IP routing

Routing is the process of selecting the best path from the source node to destination node in a network. Internet protocol provides the necessary addresses and delivery mechanism for datagrams to make this possible. Section 2.1 gives an overview of IP technology.

2.1 IP protocol

Internet protocol is the essential part of the Internet layer in the TCP/IP Protocol Suite. It offers unreliable IP datagram delivery from endpoint to endpoint, where endpoints are defined by the IP addressing scheme. The universality of the IP protocol comes from the fact that it contains only the necessary functions to deliver packets through network and fragment packets. It does not provide any flow or error control, sequencing or end-to-end data reliability. The basic IP traffic is connectionless, best-effort and there are no retransmissions.

As mentioned, the IP Protocol lies in the Internet layer of the TCP/IP protocol suite defined in Internet Engineering Task Force's (IETF's) document [9]. Table 2 illustrates the location of the IP protocol in the protocol hierarchy.

Table 2: IP Protocol Stack

Application layer				
Transport layer				
UDP	TCP		MPTCP	RSVP
Internet layer				
OSPF			IS-IS	
ICMP			IP	
	MPLS			
Network interface layer (link layer)				
	Data link layer			
ARP			Ethernet (control)	
	Physical layer			
Ethernet (physical)	ISDN	SONET/SDH	DSL	OTN

The lowest layer of the Internet architecture is the link layer, sometimes referred to as network interface layer to avoid confusion with the data link layer of the OSI model. The link layer is normally subdivided to the physical layer and data link layer [147]. It defines how the host or router connects to the network. The host could be a computer connected to Local Area Network (LAN) such as Ethernet or a router connected to a Wide Area Network (WAN) such as frame relay.

The second layer is the Internet layer that contains the IP protocol. Another important protocol, called Internet Control Message Protocol (ICMP), is also in this

layer. The ICMP closely relates to the IP protocol and it is introduced in section 2.1.3.

The purpose of the third layer, known as transport layer, is to establish and manage end-to-end communication flows and to provide a reliable service. Well known protocols in this layer are the Transport Control Protocol (TCP) and the User Datagram Protocol (UDP).

The top level of the TCP/IP stack is the application layer. It consists of application programs, that utilizes lower layers of the TCP/IP suite to achieve endpoint-to-endpoint and process-to-process communications.

In this thesis the focus is on protocols that belong to the IP layer of the TCP/IP suite. Two link-state routing protocols, IS-IS and OSPF are covered in Chapter 3. Even though IS-IS protocol was not originally Internet standard, more like it belongs to OSI model layer 2, it is considered to operate on the Internet layer like the OSPF.

The Open System Interconnection Reference Model (OSI Reference Model or OSI Model) provides an abstract framework for communication between computers as does TCP/IP protocol suite. It was developed by International Organization for Standardization (ISO) and it is documented in Telecommunication Standardization Sector of International Telecommunication Union's (ITU-T's) standard [143]. Seven layers of the OSI model and four layers of the TCP/IP model correlate quite well, but there are also differences. The OSI model is more strict in separating layers whereas from the TCP/IP point of view, layering does not have conceptual or structuring advantages and is considered harmful, as mentioned in [8].

In the OSI model, protocols handling the same functions as the IP, ICMP and ARP are the Connectionless Network Service (CLNS), the Connectionless Network Protocol (CLNP) and the End System to Intermediate System (ES-IS). The IS-IS is the only OSI reference model protocol discussed in this thesis. It is a layer 2 protocol in the OSI model and it is discussed in section 3.3.

2.1.1 IP addresses

IP version 4(IPv4) is defined in IETF's document [6]. Each device's interface connected to the Internet has its own 32-bit address. The Address can be divided into three parts; network, subnet and host. The Subnet is a collection of network addresses that have the same common address prefix. There are different kinds of fixed address formats available: A, B, C, D and E. Ever since this addressing architecture was introduced in 1981, the limitations of fixed address formats have been increasingly relaxed. Because increasing IPv4 address shortage especially with address class B, described in [2], classless inter-domain routing (CIDR), variable length subnet mask (VLSM) and network address translation (NAT) have been developed. The eventual solution for IPv4 address exhaustion is the IP version 6 (IPv6). [142]

2.1.2 IP header

The IP header is located before the data in an IP packet. The structure of the IP header is shown in Table 3. Time to live (TTL) is a very important field for routing algorithms. It defines the maximum number of links, which the packet may be routed over. Each router decreases the number at least by one. The TTL value is used to prevent accidental routing loops. Most of the fields are self-explanatory. The options and type of service fields are not used normally.

Table 3: Internet Header Format

Version	IHL	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source Address				
Destination Address				
Options	Padding			

2.1.3 ICMP

The Internet Control Message Protocol (ICMP), specified in [5], works in the IP layer. The purpose of ICMP is to provide information about errors in IP datagrams or other information about diagnostics and routing. ICMP together with the IP protocol and Address Resolution Protocol (ARP) enables the basic packet sending for hosts. ARP, defined in [7], simply translates an IP address to a physical Ethernet Media Access Control (MAC) address.

When the sending host and receiving host are not in the same subnet, discovery of the local router is needed. This is one of the tasks that can be performed using ICMP messages. [32] One of the most common ICMP messages is the destination unreachable message. It can be created in several situations, for example when the gateway's port is not active or when IP packet is too large and needs to be fragmented but the fragmentation bit of the field in the IP packet is not set.

The ICMP protocol message is not reliable because it is encapsulated inside a single IP packet. Furthermore, an ICMP message cannot be created by another ICMP error message in order to avoid infinite regress of messages.

As mentioned, IP, ARP and ICMP are enough to enable sending of IP packets between hosts inside a subnet or between a host and a local router in a LAN, but in order to transmit packets through the public Internet, information exchange between routers in different network hierarchies needs to be established. Protocols solving this problem are described in next sections. [142], [147], [145]

2.2 The purpose of router

The current Internet infrastructure consists of an interconnected set of networks. A router, also known as a gateway, is a device that connects these separate networks together. The router's main tasks can be divided into three different processes: maintaining the routing table, discovering paths to various destinations and forwarding IP packets inside the network or between different networks. Each of these tasks is explained in sections 2.2.1 and 2.2.2 respectively.

There are also other functions that an IPv4 router has to fulfill according to [4]. These functions are:

- Network interfacing
- IP processing
- network congestion and admission control
- network security and access control
- network configuration, monitoring and administration

Additional requirements for a router depend on its type. Naturally large backbone routers have different demands than small access routers.

2.2.1 Forwarding

When a router receives an IP packet at one of its incoming (ingress) interfaces, the header of the received IP packet is checked. When the destination address of the packet is known, the forwarding table lookup is performed to obtain the information to which outgoing (egress) interface the packet should be sent. There could be also several egress interfaces where the same packet is sent in case of multicast delivery.

This packet flow through the router is called forwarding. Normally the process is implemented totally on hardware or using a network processor unit (NPU) in order to achieve high speed packet transfer demands.

2.2.2 Routing database

The routing database or so-called routing information base (RIB), contains all information about the destinations known to the router. Packets are delivered based on information of forwarding information base (FIB), so it should always contain the newest information about the network.

The FIB contains only the best routes of the RIB. The selection of which routes in the routing database are installed to the FIB depends on the distance and possible metrics of the route. Administrative distance emphasizes that the distance is configured manually by the network administrator instead of the routing protocol. A

table containing different types of routes is shown in Table 4. Most vendors use these distance values. The directly connected route has the lowest cost and hence, is the best route, as the table shows. Routing algorithms can use many different metrics to determine the best route. This metric can be hop count, reliability of the route, delay, bandwidth, throughput, Maximum Transmission Unit (MTU), load or communication cost.

Table 4: Administrative Distances of Routing Protocols

Protocol	distance
Directly connected route	0
Static route out an interface	1
Static route to next-hop address	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP v1 and v2	120
EGP	140
On Demand Routing (ODR)	160
External EIGRP	170
Internal BGP	200
DHCP-learned	254
Unknown	255

2.3 Routing protocols

Normally a router is running different kinds of routing protocols that try to keep the routing database up-to-date. Routing protocols work automatically, so therefore these protocols fall in the category of dynamic routing. Conversely, static routes have to be set up manually using, for example, command line interface (CLI) or management software.

2.3.1 Routing hierarchy

Conceptually, the Internet consists of intra-domain and inter-domain (or extra-domain) networks. Protocols running in intra-domain are called Interior Gateway Protocols (IGP). They exchange routing information within a single autonomous system (AS), which has a common routing policy and single network administration. In contrast, Exterior Gateway Protocols (EGP) exchange information between an

Autonomous Systems comprizing the global Internet. The Border Gateway Protocol version 4(BGP-4) is the de-facto inter-domain routing protocol of the Internet. Nevertheless, in this thesis the focus is on IP routing in intra-domain networks. More information about inter-domain routing is available, for instance in IETF's documents [1] and [3].

2.3.2 Intra-domain routing

Traditionally intra-domain routing protocols are categorized into link-state and distance vector protocols. Distance vector protocols are based on distributed Belman-Ford algorithm described in [136] and [141]. Each router that is running the algorithm, sends periodically a list of routes to its neighbor routers. The list consists of destination, called a vector, and distance measured in hops, i.e. the number of routers to that destination. Distance vector algorithms are simple, efficient, easy to implement and they do not need almost any configuration. They solve the shortest path problem correctly to all destinations in polynomial time, but if routes change during convergence, the computation will not necessarily stabilize. Distance vector algorithms are also prone to routing loops. [145], [142], [132]

Nowadays link-state routing protocols are the basic technique in intra-domain networks. They are based on Dijkstra's shortest path algorithm that calculates the shortest path from a source node to all the destinations using a single link metric as a parameter. In contrast to distance vector protocols, link-state routing protocols are more complex but they reduce overall broadcast traffic, provide greater flexibility and make better routing decisions using more sophisticated methods by taking different link metrics into consideration. Because all routers calculate the routing paths and the information is flooded quickly to the whole network, all routers have the same consistent information of the topology of the autonomous system. Link-state routing protocols provide 2-level hierarchy for scaling purposes. These topics and also other concepts about link-state routing protocols are described more thoroughly in Chapter 3. [145], [142], [132]

2.4 Challenges in IP routing

This section introduces the most important challenges in IP networks. Routing loops, protocol convergence, congestion and route flapping problems are briefly described.

According to [15], congestion is the most significant problem in IP networks. Congestion is a state of the network resource in which the traffic incident on the resource exceeds its output capacity over an interval of time [15]. The management of congestion can be quite different when various time scales are observed. With long scale congestion (from weeks to months), investments for expand network capacity are maybe the only way to solve the problem. However, if only part of the network is congested, traffic can be distributed more evenly across the whole network by adjust-

ing IGP or BGP parameters, changing logical topology more closely to actual traffic distribution of the network or by using path-oriented technologies, such as explicitly routed labels switched paths (ER-LSPs) in Multiprotocol Label Switching (MPLS) networks. MPLS is introduced briefly in Chapter 4. The same technologies apply well in medium time scale congestion (from minutes to days), but a measurement system for monitoring the state of the network is needed to offer feedback to enable these technologies to work adaptively. Short term (from pico seconds to minutes) solutions are mostly packet level processing and queue handling functions such as Random Early Detection (RED) and the Transmission Control Protocol (TCP).

Convergence time is the feature of a protocol running in a network with certain topology and configuration. When a link failure occurs, the transmitting gateway receives information of the link failure from the routing protocol, the router starts to update its routing table and announces the new state of the link to other routers. The information spreads over the whole network and, in the end, the whole traffic is transferred through other links. This time period while the network is changing its state is called convergence time. It can be divided into four different components:

- Link failure detection time T_{det}
- Information spreading (flooding in link-state protocols) T_{flo}
- Routing calculations (shortest path calculation in link-state protocols) T_{SPF}
- Route installation to RIB and FIB of all routers in the network T_{fib}

Using mathematical formulation, the convergence time (T_{con}) is:

$$T_{con} = T_{det} + T_{flo} + T_{spf} + T_{fib} \tag{1}$$

Routing loops are one of the basic problems in IP routing. A loop can easily occur during convergence in networks, for example, when old route information exists in the routing table. Loops always spend unnecessarily bandwidth and increase the congestion in the network.

Route flapping occurs, when routing tables are automatically updated. Basically, route flapping is oscillation of the route from one path to another, which can lead to poor network performance. [15]

2.4.1 Loop avoidance and convergence problems with DV protocols

Distance vector protocols are unable to detect forwarding loops. In consequence, the maximum hop limit (TTL value) must be quite a small value, for example in the Routing Information Protocol (RIP, [37]) the limit is 16. This unfortunately limits also the size of the network. If datagrams continue to be forwarded after the TTL value of 16 has been reached, the router trashes the datagram. This counting to infinity problem occurs quite easily when the network breaks into different islands.

There are different kinds of techniques, which try to minimize the impact of slow convergence to the routing table. One such technique is split horizon, in which case the router sends new routing information to all neighbor routers except to the interface, from which the new route information arrived. Split horizon with poison reverse, however, improves the algorithm slightly. It advertises that link, from which the failed route was learned, to infinity. In this way the loops of the two routers are prevented, but larger loops are still possible.

Triggered updates is the technique that speeds up the spreading of new information to the network. Normally new advertisements are sent after every 30 second, but with triggered update, new information is sent immediately.

Also the so-called hold down is one technique to solve the slow convergence problem. The purpose of hold down is to wait long enough, commonly about 60 seconds, when some part of the network becomes unreachable. This way a destination unreachable message spreads to every router in networks and messages, that are out off date, are avoided.

RIP2 solves some of the convergence problems, because it includes the origin of each route in the update messages. [145], [142]

2.4.2 Routing challenges with link-state protocols

In link-state networks, the convergence time is always much smaller than in DV protocols. Loops are only temporary and methods that prevent the occurrence of temporary loops during convergence have been developed [56], [57]. Although the improvements made to DV protocols, link-state routing protocols provide scalability, fast convergence and much more flexible traffic engineering methods, which helps to improve network's performance. Links-state routing is the topic of Chapter 3.

3 Link-state routing

A link-state routing protocol runs in a single autonomous system. The AS can also be called the administrative domain (AD) because all routers are under the same operational administration. A router running the link-state routing protocol sends link-state advertisements (LSA) to other routers. Each router builds a topology called a link-state database of the network based on information of the received LSAs and gives it as an input to the link-state algorithm such as Shortest Path First (SPF). The algorithm computes the shortest paths in the network. The shortest paths are then used in the building of the routing table for the router. The shortest path algorithm developed by Dijkstra is the topic of the following sections.

Additionally, the two most famous link-state routing protocols are introduced in this chapter: Open Shortest Path First (OSPF) and the Intermediate System-to-Intermediate System (IS-IS) Protocol. The important issues of the two protocols are the basic functions and processes that affect the convergence time. These processes are neighbor detection, database synchronization and information flooding. The calculation of shortest paths affects the convergence time that can be decreased by fine tuning these mentioned processes. IGP fast convergence is the topic of the last section in this chapter.

Link-state routing protocols function a somewhat differently depending on the network type, which is explained in sections 3.2 and 3.3. These different types are a point-to-point network, a broadcast network and a non-broadcast multiple access (NBMA) network. The point-to-point network consists of many connections between individual pairs of nodes. Each link has only two endpoints and a traffic has to go often via many intermediate nodes to reach the final destination in the point-to-point network. The broadcast network has a single communication channel that is shared by all the nodes on the network. Many of the current local area network (LAN) architectures, such as Ethernet, Token Ring and FDDI, support directly a broadcast mechanism. In the non-broadcast multiple access network a one node is connected to another over a virtual circuit or across a switching device. Frame Relay, Asynchronous Transfer Mode (ATM) and X.25 networks are the examples of typical NBMA networks.

3.1 Dijkstra's shortest path algorithm

The shortest path algorithm was developed by Edsger Dijkstra in 1956, [139]. It is the most famous algorithm for finding the shortest path, also known as Graph Geodesic, between two graph vertexes (nodes) (u, ν) of a weighted, directed graph $G = (V, E)$. According to [146], the shortest path problem is defined as:

$$w(p) = \sum_{i=1}^{k} w(\nu_{i-1}, \nu_i), \tag{2}$$

where the weight $w(p)$ of path $p = \langle \nu_0, \nu_1, ..., \nu_k \rangle$ is the sum of the weights of its constituent edges. This sum is minimized finding the shortest-path weight $\delta(u, \nu)$:

$$\delta(u,\nu) = \begin{cases} min\{w(p) : u \rightsquigarrow^p \nu\} & \text{if there is a path from } u \text{ to } \nu, \\ \infty & \text{otherwise.} \end{cases} \quad (3)$$

A greedy algorithm makes locally optimal choices in each selection state of the running algorithm. Dijkstra's algorithm is an efficient greedy algorithm that solves optimally the single-source shortest path problem when all edge weights are non-negative. The pseudocode approach shown in algorithm 1 follows partially the book of Cormen [146].

- **S** is the set of settled nodes, the tree database in other words. The nodes whose shortest distances from the source have been found. It contains the nodes in the shortest path in the end.
- **Q** is the set of nodes unexamined, organized as candidate database in other words, normally minimum priority queue.
- **d** is an estimate of the shortest distance from source to each vertex
- **w** is the weight of the edge between u and ν (that is, $w[u, \nu]$), the link's cost in other words.
- **p** stores the predecessor of each node on the shortest path from the source.
- **EXTRACT-MIN** operation deletes the minimum element of data structure and returns it.
- **DECREASE_KEY**[ν,**Q**] operation updates the key of element ν within min-heap Q and updates the heap if the heap property becomes violated.

The verbal description of the algorithm's behavior constructing the shortest path tree (SPT) is following:

- Steps 1-5 Initialization

  The starting node is the router itself. It is the root of the tree database. The cost to itself is zero. Distance to all other nodes is infinity.

- Step 6

  All nodes described in the link-state database are added to the candidate database Q.

- Step 7

  While-loop starts. If the cost of the first path is infinity, then the algorithm terminates.

Algorithm 1 Pseudocode of Dijkstra's algorithm

```
1: for each ν ∈ G.V do
2:    d[ν]=inf
3:    p[ν]=NULL
4: end for
5: S = NULL
6: Q = G.V
7: while Q and w[Q[1] ≠ inf do
8:    u = EXTRACT-MIN(Q)
9:    S = S ∪ {u}
10:   for each ν adjacent to u do
11:      if  d[ν] > d[u] + w[u,ν] then
12:         d[ν] = d[u] + w[u, ν]
13:         p[ν] = u
14:         DECREASE_KEY[ν, Q]
15:      end if
16:   end for
17: end while
```

- Steps 8-9

  Root's neighbor u, that has the smallest cost, is removed from the candidate database Q and added to the tree database S.

- Step 10

  The loop goes through each adjacent vertex of node u

- Steps 11-14

  The current estimate (d[ν]) is compared to the sum of u's distance from the root (d[u]) and the weight of the edge between u and ν (w[u, ν]). If the sum is less than the current estimate, the sum value becomes a new estimate and the corresponding vertex ν is updated in the candidate database. This process is called relaxation, where the edge between the two vertices is relaxed.

- Steps 15-17

  If there are still items in the candidate database, it picks a new node u from the candidate database and starts examining neighbors. Otherwise the shortest path is calculated, d[ν] contains the shortest distance and S contains the nodes of the SPT.

Dijkstra's algorithm uses a greedy strategy, because the algorithm selects always the vertex having the smallest distance and it travels towards that direction and comes back to the vertex having next smallest distance, if the smaller distance was not found in the first direction. [135], [146]

3.1.1 The complexity of Dijkstra's algorithm

An upper bound of the running time of Dijkstra's algorithm on a graph with edges E and vertexes V can be expressed as a function of $|E|$ and $|V|$ using the Big-O notation.

The fastest well-known implementation of a priority queue uses a Fibonacci heap [140]. Fibonacci heap with n elements takes $O(1)$ of time in insertion and DECREACE_KEY operations. The complexity of the EXTRACT-MIN operation is $O(log(n))$. Even faster implementations exist such as [138], but the complexity of data structures is not an obvious major determinant anymore, because the number of routers is not normally that high in an AS, the CPU power of modern routers is fairly high and the hierarchical topologies of link-state protocols reduce the size of data structures.

There are E number of DECREACE_KEY operations and V number of EXTRACT-MIN operations in the algorithm. Thus, the computation cost of Dijkstra's algorithm using a Fibonacci heap is $O(|E| + |V|log|V|)$. [135], [146]

3.1.2 Dijkstra's algorithm with multiple equal cost paths

This section introduces an algorithm that finds all alternative equally optimum paths in a graph. The new algorithm is a slight modification of Dijkstra's algorithm explained in the previous section. Actually, only the step 11 of Dijkstra's algorithm needs to be modified. The modified line is in step 11 of algorithm 1:
IF d[ν] $\geq$ d[u] + w[u,ν]
Now all equal cost paths to the same node are updated to the candidate database. If the equal cost paths are also the shortest paths, they are added to the tree database. After the algorithm has stopped, the routing database contains multiple paths to the same destination node. The routers' forwarding processes can then utilize this enhancement spreading traffic across all equal cost paths.

The ECMP does not increase much complexity to the basic algorithm. Basically steps 12, 13 and 14 are performed more often. The worst case performance is the same as calculated earlier, but average complexity increases slightly depending on the network topology and the number of equal cost paths. The worst case is actually the best case from a load balancing point of view, since all nodes in the network take part in the routing and several paths exist from the source to the destination.

3.2 Open Shortest Path First

Open Shortest Path First is the most famous intra-domain link-state routing protocol that distributes routing information inside a single autonomous system (AS). The Shortest Path First (SPF) routing algorithm is the basis for the OSPF routing protocol operations to calculate the shortest path between the source and the destination. The basic operations of OSPF version 2 are specified in [10]. Several other

extensions exist but they are not introduced here. The focus is given in [10] and only upgrades relevant in the context of this thesis are described in later sections.

A typical OSPF network consists of groups of areas. An area contains routers and host devices. All routers that belong to the same area have an identical link-state database. OSPF quickly detects topological changes and calculates new loop-free routes. The distribution of a router's new local state throughout the area is called flooding. The hello protocol establishes and maintains connections to neighbor routers and selects the Designated Router (DR) and backup DR for the broadcast and non-broadcast multiple access (NBMA) networks. A designated router establish adjacencies with all routers in the area, thus participating in the synchronizing of the link-state database. The DR also originates network link advertisements on behalf of the area. The exchange protocol handles initial synchronization between the router's link-state database and the designated router. The three sub-protocols of OSPF, the hello protocol, exchange protocol and flooding protocol, are described in detail in sections of this chapter.

All OSPF protocol exchanges can be authenticated. This means that only trusted routers can participate in the routing. There are two authentication methods available. These are simple password authentication and message digest authentication (MD-5).

Routers that connect areas are Area Border Routers (ABRs) and they must be part of the AS backbone. Each ABR announces reachability data from one area to other areas. Routes learned from inter domain routing protocols, such as BGP, can be advertised throughout the AS depending on the configuration, which type of routes are allowed, although this externally derived data is kept separate from the link-state data of the OSPF protocol. Hierarchical Routing with OSPF is explained in more detail later in this chapter.

3.2.1 OSPF message header

OSPF uses five different types of messages to communicate both link-state and general information between routers within an autonomous system or within an area. These are:

- 1: Hello
- 2: Database description
- 3: Link-state request
- 4: Link-state update
- 5: Link-state acknowledgment

OSPF runs directly on top of IP and its protocol type number is 89. All OSPF packets have a common 24 byte header, which is shown in Table 5. The router ID

can be configured to any unique number in the network. But if the router ID is not configured, it is the IP address of the router by default. In cases, when the router ID is not configured but loopbacks exist, the highest loopback address becomes the router ID. Loopback is a router's virtual network interface, that is used for testing and management purposes. A router can have several loopbacks configured.

Table 5: OSPF Common Header Format

Version Number	Type=3	Packet Length
Router ID		
Area ID		
Checksum		AuType
Authentication		
Authentication		

3.2.2 Link-state advertisements

Each link-state advertisement has a common header of 20 bytes, and then there are a number of additional fields that describe the link. Three fields in the header LS type, Link-state ID and Advertising router, identify the LSA. The common LSA header format is illustrated in Table 6.

When an area's link-state database is up-to-date, each router has the same LSAs in its database relating to that area.

Even though every router has the instance of the same LSA, age information is more likely to be different in different routers. When a new update packet is received, determination of the newer LSA depends on three fields. These three fields are LS age, LS sequence number and LS checksum. At first, the newer LS sequence number determines the newer LSA. If sequence numbers are the same, the instance having larger LS checksum is considered as a newer LSA. If the checksums are the same, the LSA having the LS age set to MaxAge, is newer. Otherwise, if the difference of LS age is larger than MaxAgeDiff, the LSA having smaller age wins. Otherwise the LSAs are identical.

Table 6: OSPF Common LSA Header Format

LS Age	Options	LS Type
Link-State ID		
Advertising Router		
LS Sequence Number		
LS Checksum	Length	

The age field is the number of seconds elapsed since the LSA was created. The maximum age (MaxAge) of the LSA is 3600 seconds and the refresh time is 1800

seconds. After 1800 seconds, the originating router floods a new instance of LSA with an LS age of 0 and LS sequence number of the old sequence number plus one. If the originating router does not communicate, the value of LS age reaches MaxAge and the LSA must be removed from the database. When the first router reaches MaxAge inside the area, it refloods new information to the area that all routers remove the old LSA from their databases. Premature aging is a procedure to remove an LSA before MaxAge is elapsed. Simply setting LS Age to MaxAge and reflooding the LSA is enough. It is possible to remove only self-originated LSAs this way.

Different types of LSAs exist:

- type=1: Router-LSA:

  Defines state, cost and type of the link to the neighbor. The originating router's ID is found this way. Router-LSAs are flooded throughout the area. In point-to-point link, also IP prefix is included.

- type=2: Network-LSA

  The designated router describes the pseudonode to other routers using this LSA. It defines subnet mask of the links to router LSAs in the network and number of routers attached to it.

- type=3: Summary-LSA (IP Network)

  Defines a destination outside an area but inside an OSPF domain. The summary of one area is flooded into other areas and vice versa.

- type=4: Summary-LSA (ASBR):

  Defines routes to ASBR router throughout an OSPF domain.

- type=5: AS-External-LSA:

  Defines external routes to outside the autonomous system.

Checksum of the LSA is used for data protection of memory corruption or protection of message data corruption during flooding. LS Age is not included in checksum. Checksum is checked in three different occasions; When LSA is received, when LSA is generated and after every CheckAge interval, that is 10 minutes by default.

3.2.3 Hello protocol

As mentioned, the hello protocol has two tasks to perform. It checks that links are operational and it elects the Designated Router (DR) and backup DR. The OSPF router discovers neighbors sending periodically hello packets to all its interfaces. The default value of a period is 10 seconds, but it can be adjusted by the administrator. All neighbors, that receive a hello message, send messages back to the router and the

neighbor list of the router is updated. The router detects a link failure, when a dead interval of the neighbor has elapsed. If the router has not received a hello message from a neighbor within the dead interval, that router is removed from the neighbor list. The value of this dead interval is configurable and it is 40 seconds by default. Normally link failure is detected by the data-link protocol much earlier than dead interval. The time detecting the link failure has a major impact on convergence time of the OSPF protocol.

The Designated Router and backup DR are elected in broadcast and NBMA networks. The election process uses the priority field of hello protocol's header. At first, only routers, that have declared themselves DR are taking part in the DR election. If the priorities of the two routers are equal, the second election criterion is the highest router ID. The backup DR election goes a similar way. If none of the routers are proposed as backup DRs, the router having the largest ID is elected. If none of the routers are proposed as DRs, the backup DR becomes DR, and backup DR election process is performed again.

When the protocol starts running, the first OSPF router on the IP subnet always becomes the DR. When a second router is added, it becomes the backup DR. A router, that has the priority of 0, cannot become a DR or backup DR. If a new router is added to the area, it always accepts the existing DR regardless of its router priority. This way the new election is not performed too often. The purpose of the DR is to reduce the amount of flooding on multiaccess media. When the DR is elected, neighboring routers that are not adjacent, send flooding messages through the DR, not directly to each other. More exactly, the multicasting procedure to reduce the amount of flooding is the following: All routers flood their link-state databases to the DR, and the DR then floods that information back to other routers on that segment. When the DR crashes, the up-to-date backup DR takes the lead. The hello protocol's packet format is described in Table 7. [128]

3.2.4 Exchange protocol

The exchange protocol synchronizes databases initially. After that flooding protocol maintains the synchronization. Only in that case the shortest path calculations are correct and loop-free routing is ensured. The exchange protocol initially synchronizes the router's link database with the designated router and the backup DR in the broadcast and NBMA networks. The flooding protocol, that is described in the next section, then ensures that all databases are in synchronization.

The following description explains the behavior of the synchronization process. After the router has found a new neighbor and has established bi-directional communication with it using the hello protocol, the router starts to synchronize its database with the neighbor's database by sending database description packets to the neighbor. Before starting the real transmission, the master and the slave have to be selected.

When the router wants to start the exchange procedure, it sets the I (Initialize) and

Table 7: OSPF Hello Header Format

Version Number	Type	Packet Length
Router ID		
Area ID		
Checksum	AuType	
Authentication		
Authentication		
Network Mask		
Hello Interval	Options	Router Priority
RouterDeadInterval		
Designated Router		
Backup Designated Router		
Neighbor		
.		
.		
.		
.		
Neighbor		

M (More) bits of the packet's header to the value 1. The router also sets the MS (Master/Slave) bit on, but if the other router also wants to be the master, the final decision depends on whichever has the larger router ID. The router ID is the address of the router and it can easily be compared from messages sent by routers. The DD sequence number is set to an arbitrary value. The first message is otherwise empty.

When the master is selected, the asymmetric exchange begins. The master start to send database description packets that contain only headers of its LSAs in the database. In the following database description messages sent by the master I bit is off, the MS-bit is on and the M-bit is on. In the last message, the M-bit is off. After each packet received from the master, the slave will send an acknowledgment. The master retransmits packets, if it has not received an acknowledgment within RmxInterval. The DD sequence number is incremented by one after every received acknowledgment from the slave. The acknowledgment packet contains link-state headers of the slave's own database. MS-bit in the acknowledgment packet's header is zero.

If the M-bit of the last message sent by master is off and the M-bit of the corresponding acknowledgment message sent by the slave is on, the slave will continue to send its database descriptions to the master. The master continues sending empty description packets until the slave has finished its sending of descriptions. Now both routers know each others LSAs and which of the LSAs should be requested from the other. The master sends then a link-state request packet to the slave and the slave answers sending a link-state update packet. The master removes the particular requested LSA from the list of records to request and starts sending new link-state request packets until the list of records is empty. Now the slave sends a link-state

request packet to the master and the process continues in a similar way. After the slave has received all link-state update packets from the master, databases are synchronized, in other words, full adjacency is established between the routers.

In broadcast and NBMA networks, routers synchronize their databases only with the DR and the DR establishes adjacencies with all other routers in the area. DR uses multicast address of 224.0.0.5 called AllSPFRouters and other routers use a destination address of 224.0.0.6 called AllDRouters.

A database description packet is shown in Table 8. The packet's interface MTU field defines the size of the largest message that can be sent on this router's interface without fragmentation. The link-state request packet format that is shown in Figure 9 contains one or more LSAs. The link-state acknowledgment packet format is in Figure 10. LSA header-field contains link-state advertisement headers, that identify the LSAs acknowledged. [128], [142]

Table 8: OSPF Database Description Packet Format

20-byte-OSPF header									
Interface MTU	Options	0	0	0	0	0	1	M	MS
DBD Sequence Number									
LSA Headers									

Table 9: OSPF Link-State Request Packet Format

20-byte-OSPF header
LS Type
Link-State ID
Advertising Router
LSA Headers . . . . .

3.2.5 Flooding protocol

The flooding protocol distributes information about topology changes to all routers in the area using link-state update packets. When a router detects that the state of one of its links has changed, it sends a link-state update packet to all its neighbors

Table 10: OSPF Link-State Acknowledgment Packet Format

20-byte-OSPF header
LSA Header(s)
.
.
.
.

or to the DR. An update message also can contain several LSAs, if the state of several links have changed. The format of the links-state update packet is the same as link-state request packet shown in Table 9.

When one of the neighbors receives the update packet, it performs database lookup and tries to find the LSA from the database. If the LSA is found, the router checks that the received LSA is newer than its own LSA. If the received LSA is newer or if there is not the same LSA in the router's database, router updates its database, sends an acknowledgment back to the sending router and floods link-state update messages to all its interfaces inside the area except the link, from which the update message was originally received. As Table 10 shows, a router can collect several acknowledgments to the same packet before sending it back. Acknowledgments can be collected to the same packet even if they are generated by different neighbors. In that case acknowledgments are sent by multicasting.

To prevent unnecessary flooding, the new update packet is ignored, if the same LSA is in the database and the difference of their arrival times is less than the MinLSArrival.

The flooding procedure iterates until all routers in the area have the updated LSA and all acknowledgments are received. Because it is crucial to have an identical link-state database in an area, the reliability of the flooding protocol is as important as data delivery. Aging, sequencing, checksums, acknowledging and demanded bi-directional communication before flooding are the methods to guarantee reliability.

The DR generates the network LSA. If the DR does not exist, there will be no network LSA. The network LSA describes all the routers attached to the network. This LSA is flooded in the area that contains the network.

The packet format for the network LSA is show in Table 11. [128]

3.2.6 Hierarchical routing in OSPF

OSPF supports two-level hierarchical routing. Inside the area all routers have the same database information, but routers, that are outside the area, do not have any topology knowledge of that area. This decreases flooding traffic, the size of the link database, used memory, required processing power and improves OSPF scaling overall.

An autonomous system is split into areas and all communication between the areas goes through area 0 (or 0.0.0.0) or the so-called backbone. Area border routers (ABRs) transfer traffic between areas. All ABRs are included in the backbone and they are running the same number of copies of the basic algorithm as the number of areas they are connected to. Otherwise the backbone is the same kind of area as any other area. It can have normal internal routers that are running only one copy of the algorithm, and they are only connected to area 0.

Every ABR has the complete topology information of the backbone and all other areas ABR is attached to. An ABR builds a summary of each area and advertises the information to other ABR routers through the backbone using type 3 summary LSAs. This way every ABR has also summary information of every area. An ABR calculates all inter-area routes using all this information.

Only intra-area routes are advertised to the backbone. Inter-area and intra-area information is advertised to other areas. Summary LSAs have the packet format shown in Table 12. In type 3 LSAs, the network mask field of the packet indicates the destination network's IP address mask.

AS boundary routers (ASBR) transfer traffic between autonomous systems. Every router in the AS know the paths to all ASBRs. ASBRs do not have to be attached to the backbone. AS-external routes received from other routing protocols, such as BGP, are advertised using AS-external LSAs. Only these LSAs are distributed to all areas. The header of the AS-external LSA is presented in Table 13.

Virtual links, Stub areas and Not-so-stubby-areas (NSSAs) provide extensions to OSPF's strict policies of using areas. Virtual links provide tunneling routing information through areas without being physically connected to the backbone. Stub areas use the least resources of the areas. AS-external-LSAs are not distributed into stub areas and also distribution of summary-LSAs is optional. External destinations can be configured using the default routes in the stub area's ABRs. Not-so-stubby-areas (NSSAs) are an extension of stub areas. External routing information is distributed to NSSA using type 7 LSAs. An NSSA allows external routes to be flooded within the area but does not accept external routes from the other areas.

Table 11: OSPF Network LSA Packet Format

20-byte-OSPF header
Network Mask
Attached Router(s)
.
.
.
.

Table 12: OSPF Summary LSA Packet Format

20-byte-OSPF header	
Network Mask	
0	Metric
TOS	TOS metric

Table 13: OSPF External LSA Packet Format

20-byte-LSA header		
Network Mask		
E	0	Metric
Forwarding Address		
External Route Tag		
E	TOS	TOS Metric
Forwarding Address		
External Route Tag		
. . .		

3.3 Intermediate System to Intermediate System

The Intermediate System to Intermediate System (IS-IS) protocol is a link-state protocol standardized by the ISO in 1992. The purpose of IS-IS was to make possible the routing of packets using the ISO developed OSI protocol stack called the connection-less network service (CLNS). IS-IS was developed at about the same time that the IETF was developing OSPF.

Basic IS-IS operations are defined in the specification ISO document [144]. IS-IS is also an Internet standard [11]. Because IS-IS runs on layer 2 of the OSI reference model, it supports multiple protocols as shown in Table 14. To support transition from IP to OSI, Integrated IS-IS was created and it is documented in [12]. Integrated IS-IS supports the exchange of intra-domain routing information for the network that uses TCP/IP-based protocols. IS-IS can support IP and OSI protocol stacks simultaneously. IS-IS supports the same features as OSPF including hierarchical routing, encapsulation, authentication, Dijkstra's SPF calculation, rapid flooding and fast convergence. IS-IS has become popular because of its simultaneous support of IPv4 and IPv6.

Table 14: IS-IS Place in the Protocol Stack

CLNP	IPv4	IPv6	IPX
IS-IS common header			
Data Link Layer			
Physical Layer			

Some of the terms used by ISO differ slightly from the ones common in the Internet. Table 15 shows common terms and synonyms used in these different worlds.

Table 15: IS-IS Terminology

IETF/OSPF	OSI/IS-IS
Router	Intermediate System (IS)
Host	End System (ES)
Router ID (RID)	System ID (Sys ID)
MAC Address	Subnetwork Point of Attachment (SNPA)
Packet	Network Protocol Data Unit (NPDU)
Frame	Subnetwork Protocol Data Unit (SNPDU)
Link-State Advertisement (LSA)	Link-State PDU (LSP)
Autonomous System (AS)	Routing Domain
Backbone area	Level 2 (L2) Subdomain
Nonbackbone area	Level 1 (L1) Area
routing	routeing

IS-IS discovers neighbor routers and establishes adjacencies between them using IS-IS hello (IIH) PDUs. The forming of adjacencies differs depending on the media type. In point-to-point networks neighbors become adjacent, if the connection

is bidirectional and their authentication type, IS-IS types and MTUs match. IS-IS routers use link-state PDUs (LSPs) to inform the adjacency router about its database status.

In broadcast networks and non-broadcast multi-access (NBMA) networks, routers report their adjacencies to a Designated Intermediate System (DIS) that generates an additional LSP, commonly known as the pseudonode LSP. The DIS is responsible for conducting flooding over the LAN and also for maintaining synchronization.

Generally, routers flood LSPs to all adjacent neighbors except to the neighbor from which they received the LSP. All routers construct their link-state databases from these LSPs. A shortest-path tree (SPT) is calculated by each IS and the routing database is built using information of this SPT.

Network Service Access Point (NSAP) addresses and other addressing schemes are explained in the addressing section of this chapter.

In the standard, functionality is divided into two parts. These parts are subnetwork independent functions and subnetwork dependent functions. Subnetwork independent functions consist of four different processes:

- Receive process
- Update process
- Decision process
- Forwarding process

Subnetwork dependent functions are used for exchanging IS-IS PDUs over a specific subnetwork or exchanging hello PDUs to establish adjacencies.

IS-IS supports a two-level hierarchy to enable scalability in large networks. The network is split into areas and router is dedicated to be an intra-area router (level 1) or an inter-area router (level 2) or both at the same time. Each router belong to the same area, share the full routing information between each other and are aware of the intra-area topology. Section 3.3.6 describes IS-IS areas in more detail.

In order to compare IS-IS and OSPF protocols easily, the following sections describe neighbor discovery, database synchronization and information flooding in a similar way as the hello, exchange and flooding protocols were described in OSPF section. Generally, these processes are the main parts of IGPs in this thesis, and thus they are explained in greater detail. [133], [134]

3.3.1 IS-IS messages

All IS-IS messages are constructed of type-specific headers followed by type, length, value (TLV) encoded structures. The type and length fields are each one byte and they specify the type and length of the data in the value field. Because the length

field is one byte, the value field can vary from 0 to 254 bytes. TLV can also contain another TLV.

Different type of PDUs are:

- Level 1 LAN IS to IS Hello PDU
- Level 2 LAN IS to IS Hello PDU
- Point-to-Point IS to IS Hello PDU
- Level 1 link-state PDU
- Level 2 link-state PDU
- Level 1 Complete Sequence Numbers PDU
- Level 2 Complete Sequence Numbers PDU
- Level 1 Partial Sequence Numbers PDU
- Level 2 Partial Sequence Numbers PDU

Table 16: PDU Header Format

Intradomain Routeing Protocol Discriminator: 1 byte			
Length Indicator: 1 byte			
Version/Protocol ID Extension: 1 byte			
ID Length: 1 byte			
R	R	R	PDU type: 6 bits
Version: 1 byte			
Reserved			
Maximum Area Addresses: 1 byte			
PDU specific fields			
Variable-Length Fields (TLVs/CLVs)			

After the common header shown in Table 16, there is a packet specific header that is followed by optional packet data. Some of the commonly used TLVs are:

- Area Addresses
- IS Neighbors
- Protocols Supported

- IP Interface Address
- IP Internal Reachability Information
- IP External Reachability Information
- Authentication Information
- Padding

3.3.2 Address format

IS-IS uses OSI-style addresses in its PDUs. The network service access point (NSAP) address is the boundary between the network and transport layers. Each transport layer entity has a single NSAP address to define which kind of network services are provided. A network entity title (NET) is an NSAP address with an n-selector of zero. NET is used in CLNS-based networks to identify the network layer of a system without a specifying transport layer entity. In IS-IS, the NET address identifies the IS. An IS can have a single NET address or multiple NETs, if it belongs to multiple areas.

The NET address format is shown in Figure 1. A simplified format is normally used. NET consists of an area ID and a system ID. The authority and Format Identifier (AFI) fields identify the used addressing domain.

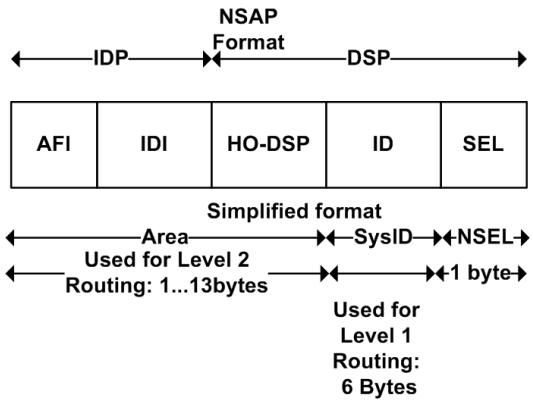

Figure 1: NSAP Address Format.

An example of creating an NET address from IP loopback, if the operator does not have an OSI address:
IP loopback address 123.23.123.3
AFI= 49 (local address domain)
area1=0001
NSEL=00

NET address: 49.0001.1230.2312.3003.00

3.3.3 Hello protocol

IS-IS exchanges hello messages to discover neighbors and to build adjacencies between them. IS-IS neighbors are adjacent after they have established bidirectional communication. The procedure of establishing adjacencies between neighbors differ depending on the network type. The two alternatives for network type are point-to-point and broadcast (LAN). In point-to-point networks, adjacency is formed, when each side of a point-to-point link declares the other side to be reachable if a hello packet is received from it. When this occurs, each side then sends a Complete Sequence Number PDU (CSNP) to trigger database synchronization. Hello PDUs are sent periodically to check the link's availability.

IS-IS three-way handshaking on point-to-point links has been specified in [13]. This extension uses a point-to-point three-way adjacency TLV. When an IS-IS router supports the three-way option, it looks for the three-way adjacency TLV in the hello messages of its neighbors. The absence of this TLV means that the neighbor does not support the option. In this case, the router reverts to the standard ISO two-way handshake. Normal behavior for IS-IS is to ignore TLVs it does not understand. Point-to-point links have only a single neighbor by definition, in which case there can be a level 1 adjacency, level 2 adjacency or both. There are no designated routers in point-to-point links.

IS-IS uses a technique similar to OSPF's designated router (DR) to build adjacencies in broadcast networks. In IS-IS, DR is called Designated Intermediate System (DIS). All other routers sharing the broadcast link synchronize their link-state databases with the DIS. Although multicast is used for communication between the DIS and other routers on the broadcast network, there is no special multicast address for the DIS. Thus adjacencies are build between every router in the LAN. The election process is relatively simple. The selection is based on the highest priority of the ISs in the level 1 sub-domain or level 2 sub-domain. If the priority values are all the same, the elected DIS is the router, whose interface connecting to the network has the highest SNPA. The router will not be the DIS as long as another router has a higher priority. Moreover, when a new router with a higher priority starts up, it will preempt the existing DIS. An IS-IS router runs this election process every time a LAN hello is received from an adjacent neighbor and every time it transmits its own LAN hello as long as there is at least one adjacent neighbor.

The format of the IS-IS LAN hello is shown in Table 17 and point-to-point hello in Table 18. LAN ID field in the LAN hello message contains the ID of the DIS and the circuit ID to differentiate LAN IDs on the same DIS.

The format of point-to-point hello is very similar. Only a couple of differences exist. There is no priority field, because the designated routers are not elected in point-to-point links. Instead of the 7-byte LAN ID field, there is a 1-byte Local Circuit

ID field that has only a local informational purpose.

Table 17: PDU LAN Hello Header Format

Intradomain Routeing Protocol Discriminator: 1 byte						
Length Indicator: 1 byte						
Version/Protocol ID Extension: 1 byte						
ID Length: 1 byte						
R	R	R	PDU type: 6 bits			
Version: 1 byte						
Reserved						
Maximum Area Addresses: 1 byte						
R	R	R	R	R	R	Circuit Type
Source ID						
Holding Time						
R	PDU Type					
Priority						
LAN ID						
Variable-Length Fields						

Table 18: PDU Point-to-point Hello Header Format

Intradomain Routeing Protocol Discriminator: 1 byte						
Length Indicator: 1 byte						
Version/Protocol ID Extension: 1 byte						
ID Length: 1 byte						
R	R	R	PDU type: 6 bits			
Version: 1 byte						
Reserved						
Maximum Area Addresses: 1 byte						
R	R	R	R	R	R	Circuit Type
Source ID						
Holding Time						
R	PDU Length					
Local Circuit ID						
Variable-Length Fields						

When an adjacency is established, the router sets the holding time for the adjacency to the same value as the holding time is in the neighbor's hello PDU. The holding time field in the hello message defines how long the IS's neighbors should wait,

before the router is declared dead. The default hold time is normally three times the hello interval, but they can also be configured separately. In broadcast networks the neighbor priority is also set according to the value of the priority field in the LAN Hello. [133], [134]

3.3.4 Database synchronization and flooding

In order to guarantee identical database information in all routers of a network, the databases have to be synchronized between neighbors and distributed to the whole network. Flooding over point-to-point links is described as reliable, because LSPs transmitted over such links must be acknowledged with a Partial Sequence Number PDU (PSNP) by the receiving router. In broadcast media, Complete Sequence Number PDUs (CSNPs) are periodically transmitted by the Level 1 DIS and the Level 2 DIS to assist other routers to synchronize their databases.

The three PDUs, shown in Tables 19, 20 and 21, are used for link-state database synchronization. These are the Link-State PDU, (LSP), the Partial Sequence Number PDU (PSNP) and the Complete Sequence Number PDU (CSNP). There is no particular acknowledgment message in IS-IS. Instead, CSNP and PSNP are used in the synchronization process. They contain descriptions of LSPs by listing their remaining lifetime, LSP ID, sequence number, and checksum fields. The difference between the two is that CSNPs contain all LSPs in the transmitting IS's link-state database, whereas PSNPs contain only a subset of the LSPs in the originator's database.

Three fields identify the LSP; source ID, pseudonode ID and LSP number. Source ID is the SysID of the originating router. Pseudonode ID identifies the pseudonode and it is originated by the DIS. The maximum size of LSP is 1492 bytes that can be exceeded, if the number of TLVs is large enough. In that case, LSP can be split into multiple parts and the LSP number identifies the part of the LSP.

In point-to-point links, ISs describe their databases by sending CSNPs to each other. If received CSNP contains an unknown LSP or more recent instance of LSP, IS sends a PSNP to the neighbor to request these LSPs. If IS has newer LSP than the received one, it sends a copy of its own LSP to the neighbor.

In broadcast links, all IS-IS routers synchronize their databases with the DIS and the DIS periodically multicasts a CSNP using the applicable multicast address, allL1IS (01-80-C2-00-00-14) or allL2IS (01-80-C2-00-00-15). There is no need for explicit acknowledgements. The CSNP describes all the LSPs in the database. Other routers then check that their databases are in synchronization. If a IS's own database does not contain all information received from the DIS, the IS requests these missing LSPs by sending out the PSNP that contain the summary of missing LSPs. The requester then receives the complete LSPs from the DIS or other peers on the link.

Two important internal flags are used in database synchronization and flooding. These are the send routing message (SRM) flag and the send sequence number (SSN) flag. The SRM flag is used by the update process to control sending of

LSPs to adjacent neighbors. The IS-IS router creates one SRM for each LSP per link. The SRM flag is set when LSP needs to be sent to a particular link. After every minimum LSP transmission interval, the link-state database is scanned and the SRM flags are checked. In point-to-point links all LSPs, that have associated with that interface (ie. SRM flag is on), are sent. The SRM flag is cleared only when that LSP is acknowledged. This means that the LSP is removed from the retransmission list. The router will continue retransmitting LSPs that have not yet been acknowledged. In broadcast links, one or more but less than 10 random LSPs are sent after scanning. This prevents the DIS receiving several of the same LSPs from different routers. The flag is cleared right after the LSP is sent.

The SSN flag is used to acknowledge received LSPs in point-to-point links and to request complete LSP information during database synchronization in broadcast links. Also similar to SRM, each LSP has one SSN flag for each link. If the SSN flag is set, the corresponding LSP should be included in the next PSNP. The SSN flag is cleared when the PSNP is sent.

Again, detailed synchronization procedures are different in point-to-point and broadcast networks. When new adjacency is established on a point-to-point link, the link's SRM flags of all LSPs are set and CSNPs are sent to each other. The received LSPs are compared to the routers' own LSPs. To be exact, the CSNP's start LSP ID, end LSP ID, sequence number and age fields are the compared factors. The following sequential comparison is performed after every receiving of a new LSP:

- If the received LSP is the same as router's own LSP, the SRM flag is cleared and the SSN flag is set in order to send a PSNP acknowledgement to the neighbor.
- If the router does not find the received LSP from its database, a new LSP entry with a sequence number of 0 is added. The SRM flag relating to this link is cleared, SRM flags relating to other links is set and the SSN flag is set in order to prevent flooding of the LSP to this link, to enable flooding to other links and to enable sending of a PSNP acknowledgement to the neighbor.
- If the received LSP is newer, the SRM flag relating to this link is cleared, SRM flags relating to other links are set and the SSN flag is set in order to prevent flooding of the LSP to this link, to enable flooding to other links and to enable sending of a PSNP acknowledgement to neighbor. The existing LSP is replaced by the new one.
- If the received LSP is older, the SSN flag is cleared and an SRM flag is set to enable sending of a newer LSP to the neighbor.
- If there is an LSP in database, which LSP ID is in the range specified by the CSNP's Start and End LSP ID fields and its sequence number is nonzero and there is remaining lifetime, but the LSP entries TLV does not contain the LSP, the SRM flag is set to enable its sending to the neighbor.

After receiving the new LSP in broadcast link, almost the same sequential comparison is performed as in point-to-point link. The exception is that the LSP's SRM flag is cleared as soon as the LSP is sent. If the new LSP is not indicated in subsequent CSNPs, the sending router resets the SRM and retransmits the LSP. After $CompleteSNPInterval$, the sending routers will receive the CSNP as an indirect acknowledgement. The other exception is that if the LSP is newer than or equal to the LSP in the database, the SSN flag is not set. In consequence, PSNPs are not sent to acknowledge these LSPs. In summary, all routers have the same knowledge of the synchronization process in a broadcast network. [133], [134]

Table 19: IS-IS Link-state PDU Format

Intradomain Routeing Protocol Discriminator: 1 byte			
Length Indicator: 1 byte			
Version/Protocol ID Extension: 1 byte			
ID Length: 1 byte			
R	R	R	PDU type: 6 bits
Version: 1 byte			
Reserved			
Maximum Area Addresses: 1 byte			
PDU Length			
Remaining Lifetime			
Source ID			
Pseudonode ID			
LSP Number			
Sequence Number			
Checksum			
P	ATT	OL	IS Type
Variable-Length Fields			

3.3.5 LSP creation and removal

This section describes the situations when a new LSP is generated and how it is removed. The following procedure is performed for determining, which of the LSPs is the newest one. If one of the LSPs has a remaining lifetime of 0, it is the most recent. If the remaining lifetimes of both LSPs are non-zero, the PDU with the larger sequence number is the most recent. If the remaining lifetimes of both LSPs are non-zero, the sequence numbers are equal and no checksum error has occurred, the LSPs are considered identical.

The LSP's remaining lifetime starts from the $MaxAge$ that is 1200 seconds by default but can be configured up to a maximum of 65,535 seconds, and counts down to zero. The IS, that originated the LSP, must periodically refresh its LSPs to

prevent the remaining lifetime from reaching 0. The refresh interval is 900 seconds plus a random value maximum of 25% from the interval value. If the remaining lifetime expires, the LSP will be kept in the database for an additional 60 seconds, known as $ZeroAgeLifetime$, before its removal.

A checksum detects corruption of the received LSP. If an LSP with an incorrect checksum is received, the remaining lifetime value is set to 0, the body of the message is removed and it is reflooded. This triggers the IS, that originated the LSP, to send a new LSP.

The LSP sequence number is always increased by one to indicate that the new LSP is sent. The value of the sequence number starts from 1 and the maximum value is 0xFFFFFFFF. If the maximum value of the sequence number is reached, IS-IS protocol must be shutdown for the time of ($MaxAge + ZeroAgeLifetime$) to let the ageing process remove the LSP from all databases. The new instance of the LSP with a sequence number of 1 is then flooded. [133], [134]

The following events generate a new LSP:

- Router startup
- The periodic refresh timer expires
- A new adjacency is established
- An adjacency or link changes state
- The metric associated with a link or reachable address changes
- The router's SysID changes

Table 20: IS-IS Complete Sequence Numbers PDU Format

Intradomain Routeing Protocol Discriminator: 1 byte			
Length Indicator: 1 byte			
Version/Protocol ID Extension: 1 byte			
ID Length: 1 byte			
R	R	R	PDU type: 6 bits
Version: 1 byte			
Reserved			
Maximum Area Addresses: 1 byte			
PDU Length			
Source ID			
Start LSP ID			
End LSP ID			
Variable-Length Fields			

- The router is elected or superseded as DIS
- An area address associated with the router is added or removed
- The overload status of the database changes
- Changes in inter-area routes
- Changes in redistributed (external) routes

3.3.6 Hierarchical routing in IS-IS

IS-IS supports a two-level hierarchy to achieve scalability in large networks. The network domain is divided into areas, which corresponds to a level 1 sub-domain. The router can operate on level 1, level 2 or both. Level 2 routers connect all areas within a routing domain. Each level 2 router has its own area address (area ID in the NSAP) that is advertised to other level 2 routers. All level 1 routers in the same area have the same area address. Topology information from an L1 routing domain is not advertised into the L2 routing domain, just reachability information. The adjacency table of all level 1, level 2 and area ID combinations is shown in 22 [133].

Usually a level 1 router does not have any knowledge of external routes. Sometimes this leads to suboptimal routing. The solution to this problem is so-called route leaking, in which a level 2/level 1 router advertises routes to level 1.

In IS-IS, it is a possible to use multiple area ID's in hello messages, which enables splitting, merging and renumbering of areas without interrupting normal operation of protocol. [133]

Table 21: IS-IS Partial Sequence Numbers PDU Format

Intradomain Routeing Protocol Discriminator: 1 byte			
Length Indicator: 1 byte			
Version/Protocol ID Extension: 1 byte			
ID Length: 1 byte			
R	R	R	PDU type: 6 bits
Version: 1 byte			
Reserved			
Maximum Area Addresses: 1 byte			
PDU Length			
Source ID			
Variable-Length Fields			

Table 22: Adjacencies with Different L1, L2 and Area ID Combinations

R1 Type	R2 Type	Area IDs	Adjacency
L1-only	L1-only	Same	L1
L1-only	L1-only	Different	None
L2-only	L2-only	Same	L2
L2-only	L2-only	Different	L2
L1-only	L2-only	Same	None
L1-only	L2-only	Different	None
L1-only	Both	Same	L1
L1-only	Both	Different	None
L2-only	Both	Same	L2
L2-only	Both	Different	L2
Both	Both	Same	L1 and L2
Both	Both	Different	L2

3.3.7 The problem with ECMP and pseudonodes

As already mentioned, a pseudonode is the DIS's representation of the multiaccess link in broadcast networks. All routers, that are connected to broadcast link, form adjacencies with this virtual node.

According to Doyle's book [133], additional behavior to Dijkstra's algorithm is needed, when an ECMP group consists of a mixture of point-to-point and broadcast links. If there is one or more links to a pseudonode and at least one link to a router in the ECMP group, link to the pseudonode should always be selected first before any link to the router. Otherwise all traffic will transfer to the point-to-point link.

3.4 IGP fast convergence

IGP fast convergence is a traffic engineering method that tries to minimize IGP's convergence time. As earlier mentioned, convergence time consists of four different components that are failure detection time, event propagation time, route calculation time and route installation time to RIB and FIB. The speed of the FIB and the RIB installation processes depend on related software and hardware implementation. Remaining portion of the convergence can be affected by several methods, the easiest of which is the fine tuning of the timers of the protocol.

If the IGP's failure detection time is not good enough after timer adjustment, it can be improved using bidirectional forwarding detection (BFD) [36]. Other methods need changes to the protocol itself. Incremental SPF (ISPF) calculation algorithms reduce route calculation time and also the CPU load. Partial Route Calculation (PRC) is the SPF for the SPT's leaf. The use of an intelligent timer means that the timer intervals can be changed dynamically by setting the initial time delay,

incremental time interval and maximum time interval. This way responses to a small number of changes are fast, but overloading is prevented in multiple events. The timer tuning is problematic, because using too small timer values would result route flapping and finding optimal timer values could be time consuming especially if network changes are not easily predictable.

Also protocol-specific improvements exist. In OSPF, immediately replaying hello avoids the waiting for the next periodic hello [43]. Optimized database description exchange reduces the synchronization overhead [44].

In IS-IS, a fully-meshed network produces lots of flooding traffic. The amount of flooding traffic can be reduced using mesh-groups defined in [14]. [133]

4 Traffic engineering

Traffic engineering (TE) a is method that jointly optimizes network efficiency and performance to the current network conditions.

Broadly speaking traffic engineering covers all network related concepts, such as network traffic measurement, analysis, modeling, characterization, simulation and control. Basically the only network related concept, that is not part of TE is network engineering, manipulates network resources and manages the network on a long time scale. The forecasting of network usage and investment in the right parts and recourses in the network is essential in cost-efficient network business. Carefully designed network planning also helps utilizing TE methods and it improves the network efficiency because network optimization becomes an easier, sometimes trivial task.

The optimization and control aspects are central to traffic engineering. Traffic engineering can be examined from different point of views as Table 23 shows.

Table 23: Traffic Engineering Classifications

Intra-domain	vs.	Inter-domain
Online	vs.	Offline
Unicast	vs.	Multicast
MPLS-TE	vs.	IP metric-based TE
Short-term	vs.	Long-term
Host-based	vs.	Network-based TE
Centralized	vs.	Distributed TE

In this thesis, the focus is on intra-domain unicast IP traffic engineering. The difference between online TE and offline TE is the time scale, when weights, traffic splitting or scheduling of the routing are adjusted. Another difference is the availability of the traffic matrix (TM) that represents each pair of traffic's ingress and egress points of the network. Naturally, this matrix changes over time and the measurement and prediction of TM are demanding tasks. There are lots of ways to find optimal weights, if TM is known. In an intra-domain, TM is slightly easier to handle, because both ingress and egress points of the matrix are fixed, contrary to the inter-domain case. Offline TE is a more popular field than online TE and it is the central part of this thesis as well, although online methods have been more heavily studied during recent years. Thus, some of the new online TE methods are introduced.

Additionally, two important concepts of traffic engineering are discussed, protection and load balancing. In a way, load balancing is a solution to congestion problem and protection is solution to failures in the network.

This chapter introduces several traffic engineering methods starting from IGP metric manipulation. The Equal Cost multipath is introduced in Section 4.3. Other

important load balancing and protection mechanisms are introduced after that.

Because MPLS-TE has become very popular in recent years, a brief introduction of this technology and a comparison between MPLS-TE and IP TE is represented. [137]

4.1 Network optimization

In general, the main challenge of network optimization is to avoid the situation where the network is congested in some links while other parts of the network are underutilized.

Networks can be optimized using different kinds of objectives. The most popular objective is minimizing maximum utilization in a normal working case. One common strategy is minimizing maximum utilization under defined failure conditions. Also other objectives exist like minimizing propagation delay, etc. The objective functions give the more detailed description of the goodness of the designed algorithm. However, all the functions do not always fulfill the objective of being of good merit, which reflects the traffic engineering goals. The authors in [74] give detailed descriptions of the most popular objective functions.

The following sections describe different kinds of TE methods.

4.1.1 Altering IGP metrics

One of the oldest TE method is to alter the link weights of the network's routing protocol and to find a better utilization of network resources this way. In offline TE, the IGP's metrics can be solved using a linear programming formulation [58]. By adding equal cost paths into the formulation, the problem becomes NP hard [85], [87]. Some of the algorithms, such as the Genetic Algorithm (GA) [88], Hybrid Genetic Algorithm [89] and Lagrangian Relaxation based approach [127] achieve near optimal results. Optimal results can be achieved using unequal load splitting and unequal cost paths. [93]

Static offline TE optimization has the major problem that these kind of weight settings do not normally respond well to dynamic traffic variations, although robustness to link failures can be taken into account at some level, when an optimization algorithm is being developed. For example, the authors in [82] propose to use l-balanced weight settings that controls the maximum utilization level in the network. The remaining part of the capacity is needed, when unpredictable changes are handled. Similar optimization algorithms exist for MPLS networks.

This TE approach is not technology dependent, so it can be applied to singlepath, multipath and MPLS cases.

4.1.2 Network optimization tools

The optimal algorithm does not help the service provider (SP). The algorithm has to be implemented in a toolbox, that is easy to use by the SP. There are several optimizing tools available, for instance the open source software called Interior Gateway Weight Optimizer, [84]. It optimizes link parameters when the topology and traffic information of the network is given. IGP-WO is part of the TOTEM toolbox [79] and it uses an algorithm developed by Fortz and Thorup [85]. Commercial optimization tools are, for example, MATE [124], Wandl IP/MPLSView [125] and Opnet SP Guru [126].

Some of the tools provide self-configuration, in which case the tool running in a centralized server solves the optimal weights, but also sends the weights to routers without manual configuration. TOTEM can be used as a online tool. Another new self-configuring tool, SculpTE, [121], identifies the most loaded link and updates its weight with its key metric in such a way, that the shortest path of this flow is transferred to another link. SculpTE requires that the link load and link weight updates are propagated throughout the network once every iteration. Altering the weights of routing protocol online is problematic, because it easily creates stability issues [81].

4.1.3 Multi topology routing

Multi topology (MT) routing is one of the most promising technologies providing robust traffic engineering also in failure scenarios. MT for IS-IS and OSPF have Internet standards defined in [53] and [55]. In MT, routers maintain several independent logical topologies allowing different kinds of traffic to be routed independently through the network. MT techniques can be used in online and offline TE, centralized and distributed TE and IP and MPLS TE. The drawback of this method is massive consumption of computation resources and memory in router. [80]

4.1.4 IGP optimization using multiple metrics

Better optimization results can be achieved, if multiple metrics are used in optimization. Although the IS-IS and OSPF standards specify four different metrics, vendors support only a single metric. Although optimized weights produce better results, this method has the same problem as all static weight settings. There is no adaptation to different traffic conditions and reconfiguration is needed after a time interval. The impact of weight ranges have also been studied. In order to benefit multiple shortest paths the most, a small ratio of maximum value/minimum value is preferred. [97], [46], [76]

4.1.5 Unequal cost routing

Several existing and proposed routing protocols are designed to use unequal cost paths to a destination. Some of them provide optimal routing. For example, penalizing exponential flow-splitting (PEFT) splits traffic over multiple paths with an exponential penalty on longer paths [93]. However, this technique requires changes to the existing IGPs and it is not able to handle traffic dynamics. Different modifications of the k-shortest paths algorithm is one alternative of finding multiple unequal cost disjoint paths to a destination. Overall, unequal cost routing algorithms require too large changes to standard IGP solutions. Additionally, due to their complexity and problems with troubleshooting and configuration, they have not become popular. Almost always more practical ECMP solutions are preferred.

4.2 Different aspects of multipath load balancing

In general, multipath load balancing refers to distributing traffic more evenly by adding somehow additional paths to the router's forwarding layer and using load balancing algorithm to identify flows and distribute them to different paths.

The multipath load balancing concept can be utilized in different layers of the IP stack. Well-known technologies, such as ECMP, link aggregation, MPLS-TE load balancing, pseudowire (PW) load balancing, Multi-Path BGP (MBGP) and subnet load balancing with Virtual Router Redundancy Protocol (VRRP) and Gateway Load Balancing Protocol (GLBP), use traffic splitting. Again, to keep the focus on intra-domain networks, MBGP is omitted here. More information of the technology is available in [47], [45] and [73].

In storage networking, protocols, such as Fibre Channel over IP (FCIP) and Internet Small Computer System Interface (iSCSI), provide their own session-based load balancing features that are utilized on the transport layer. In these technologies, load balancing is based on multiple TCP connections.[51],[52].

PW load balancing is described next very briefly. Other technologies are described later in this chapter.

4.2.1 Load balancing in pseudowires

Pseudowire (PW) is a mechanism for emulating various Layer 2 networking services such as Frame Relay, ATM, Ethernet, TDM, and SONET/SDH over packet switched networks (PSNs) such as Ethernet, IP, or MPLS. Pseudowire is a very widely used technology to allow a seamless connection between two network elements by creating logical links, or virtual tunnels, across the packet network.

PW load balancing refers normally to general load balancing mechanisms used in PWs or two competing PW specific technologies, PW bonding and Flow Aware Transport of MPLS Pseudowires. Both have IETF's documents available [48] and

[49]. PW bonding uses round robin traffic distribution and re-assembly based on sequence numbers at the receiving end.

Flow Aware Transport of MPLS Pseudowires introduces an additional label that can be used with ECMP, link aggregation and MPLS to identify and distribute subflows to different paths. A similar method for general MPLS load balancing called entropy labels, is introduced in section 4.5.4.

4.3 Equal cost multipath

Link-state protocols such as OSPF and IS-IS are based on the shortest path first (SPF) algorithm that calculates the single shortest path from a source node to a destination node. Equal Cost Multipath is a technique that enables using several equal cost paths in IP routing. This feature helps to distribute traffic more evenly in the network but it is also a protection method. With ECMP, equal cost paths are installed to the load balancing table in the forwarding layer of router, and after detection of a link failure, traffic can be distributed between the rest of the equal paths in a subsecond and without severe loss of traffic.

ECMP does not require any special configuration, because SPF computes automatically equal cost paths and these paths are then advertised to the forwarding layer. The only variable factor is the number of ECMP paths. The limiting factor is the maximum number of ECMP paths the load balancing algorithm can support. Normally number of ECMP paths can be configured between 1 and the maximum value of supported paths. Common values of maximum paths are 8 and 16.

ECMP does not take into account any differences in the bandwidths of the outgoing interfaces, but it has been a common practice to use a link metric that is proportional to the inverse of the link's bandwidth. However, it is a better way to create additional ECMP paths by adjusting metrics in an appropriate way. Fortz and Thorup reported load balancing improvement of $50\% - 110\%$ compared to singlepath case [85], [86]. Normally ECMP is implemented to link-state routing protocols, because they need quite small modification to their path calculation, but ECMP implementations to DV protocols have been published as well. For example, the Interior Gateway Routing Protocol (IGRP) and Enhanced Interior Gateway Routing Protocol (EIGRP) support ECMP.

An example of ECMP load balancing is shown in Figure 2. Traffic is spread quite evenly to the whole network. Additionally, these three ECMP paths are backups for each other. If one of the paths fail, traffic is split between the other two paths after failure detection. There is only one node and one link that shares more than one path. In this case, only the source router needs to support ECMP. Although the other routers would support only single path routing, ECMP would still work the same way. However, then only the source router could split traffic and provide protection. If a mixture of single path and ECMP routers are used, it is very important to use appropriate weight settings so that sufficient amount of ECMP paths are created.

From a protection point of view, the more ECMP paths exist, the better the scope of protection. Naturally, backup paths should be good enough to carry additional traffic. From a load balancing point of view this is not the case. Adding ECMP paths will increase the average load balancing performance, but optimal weight settings depend on hop count, bandwidth and delay of the used links. Actually, the authors in [75] proposed an optimization algorithm that optimizes load balancing, but is also robust to all single link failures. Additionally, splitting the traffic sequentially in every node of a multi-stage setup induces polarization effects and increases packet reordening and route flapping. Unequal splitting of ECMP paths and the use of anti-polarization mechanisms contributes to joint optimization of load balancing and protection. The next section, that discusses load balancing algorithms, provides more information about the mentioned mechanisms.

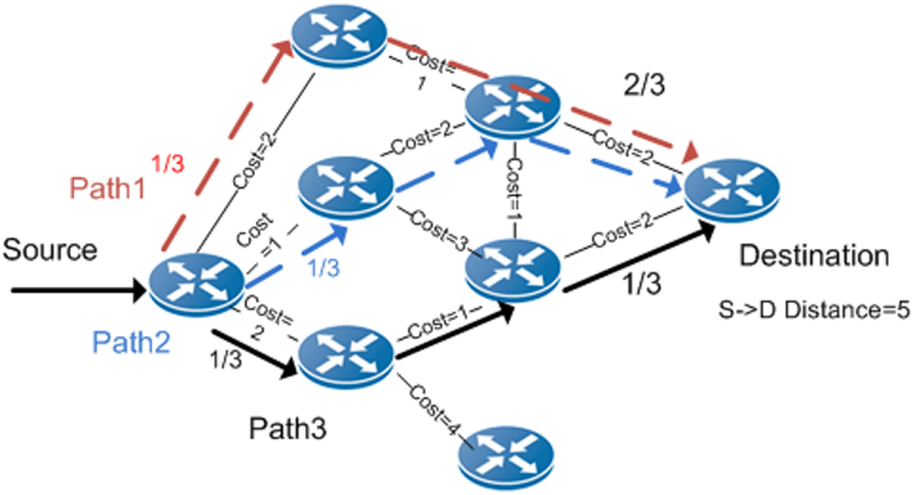

Figure 2: ECMP Load Balancing.

4.3.1 Load balancing algorithms

One implementation issue is how to distribute traffic evenly, when several equally good paths are provided by the router's control plane. Overall, the goals of a load balancing algorithm are minimizing consumption of network resources, providing load balancing and minimum interference in a simple, efficient and implementable way. Several hash-based algorithms have been developed to solve the problem. Traffic should be split using flow-based traffic distribution in order to avoid massive packet reordering, because different paths have always different delays and hence packets are delivered out-of-order. Because of this, packet-based algorithms, such as round robin, should not be used. Other smaller issues are the unpredictability of the packet's path, which would create problems with debugging tools such as Ping and Traceroute, and it would create variation in the Maximum Transmission Unit (MTU) between different paths. However, there are also drawbacks in using pure flow-based hashing. First of all, the total traffic is not distributed evenly. Now the

total traffic distribution depends on different flows. If a single flow dominates and uses a huge amount of bandwidth compared to other flows, the traffic cannot be well balanced, because packets of a single flow select always the same path.

The latest algorithms optimize jointly packet ordering and load balancing. An easy improvement is to use packet-based algorithms with UDP packets and flow-based algorithms with TCP. The division of packets between UDP and TCP and handling them differently provides better results, but sometimes also UDP packets needs to be in order, like in Voice over IP [94].

Additionally, the delivery of packets out-of-order is not as severe as it sounds. TCP's congestion window can decrease the effect of packets out-of-order. TCP detects loss of a packet or segment by finding the missing sequence number in the packet queue. The receiver sends an acknowledgement (ACK) to inform the sender that a segment was lost. When the sender has received three duplicate ACKs, it retransmits the corresponding segment and reduces the sending rate by about half. Since none of the packets are missing in reality, they were just received out-of-order, retransmissions and especially reduced sending rate has a large impact on TCP performance. IETF's document [40] specifies non-congestion robustness (NCR) for TCP, which mitigates original packet loss detection by increasing the number of duplicate acknowledgments to approximately the data size of a congestion window. The maximum size of a congestion window depends on the buffer size of the receiver, transfer delay, minimum MTU of the path and the amount of traffic in the path. In other words, the maximum window size is the maximum amount of data which can be sent without having to wait for the ACK of the transmitter. Consequently the size of the congestion window depends on the current state of the path and cannot be increased in order to facilitate packet re-ordering. However, with the help of this window, small variations of delay between different paths becomes acceptable so long as the number of consequtive packets out-of-order is smaller than the size of the TCP congestion window. Also the authors have shown in [59] that the change of packet flow does not produce many out-of-order packets, when modern jointly optimized algorithms are used. [18], [40], [41]

In addition to multipath routing, other causes of packet ordering exist. According to [77], these causes are route flapping, inherent parallelism in modern high-speed routers, link-layer retransmissions and the buffering of packets in routers during the processing of a routing update. Several measurements prove that packet reordering is not a rare phenomena [113], [114]. Consequently TCP improvements to packet reordering issue should be utilized.

Other interesting factors of algorithms are the disruption of an active flow due to nexthop addition or deletion and performance issues that are the speed of nexthop selection and the cost of implementation. Every hash algorithm tries to create a uniform distribution of flows between egress ports, therefore load balancing works more evenly when the number of flows increases. The selection of flows should not be totally predictable, because this could make the network vulnerable to denial-of service attacks. [22]

Load balancing algorithms calculate a hash over selected fields in the packet header. Normally source and destination addresses (SA and DA) of the IP header are used but also the protocol field and type of service field of the IP header, the SA and DA of the MAC layer or source and destination ports (SP and DP) can also be used. In IPv6, the IP header has its own flow identification field, so a combination of source and destination addresses and the flow label should be enough for load balancing [20], [21].

Usually paths are split equally, but also unequal cost load sharing algorithms exist. In order to achieve real benefits with unequal cost load balancing algorithms against equal cost splitting, near optimal splitting of traffic is needed. This means nearly realtime traffic measurements to give feedback and achieve adaptation. OSPF optimized multipath (OSPF-OMP), [50], is one of the first unequal traffic splitting methods. The feedback part is created using new opaque LSA, LSA_OMP_PATH_LOAD that carries the load information of the path. Simulations show that OSPF-OMP gives better results than ECMP [90], but OSPF-OMP can also cause oscillation, which introduces delay variation and packet reordering [122].

Vendors have similar approaches to implementing their load balancing algorithms. In Juniper, an older Internet Processor ASIC supports 8 load balancing paths and only packet based load balancing is possible. Newer Internet Processor II ASIC supports 16 equal-cost paths and both packet-based and flow-based load balancing are possible to configure. Also Cisco supports 16 equal-cost paths and both load balancing types are possible to use. Tellabs older BRAIN ASIC support 8 ECMP paths and the upcoming BRAIN2 16 ECMP paths and both use flow-based load balancing.

There are several flow-based algorithms available. Pure flow-based algorithms, Modulo-N Hash, Hash-Threshold and Highest Random Weight (HRW) are introduced in [22] and [23]. All algorithms keep the same flow on the same path. The major differences discussed are disruption factor and computational complexity. Disruption factor is the measurement of how many flows are changed to another path due to nexthop addition or deletion. According to the documents, HRW has the best disruption factor, but it has the highest computational complexity. Hash-Threshold has almost as good disruption factor as HRW has and it is computationally less complex. In documents [78] and [123], the authors propose Hash-Threshold (CRC16) for the best static load balancing.

Hash-Threshold or Direct Hashing (DH) calculates a hash, normally cyclic redundancy check (CRC), based on fields, which defines the flow. Fields are collected to the hashing value H that is the input of the hash function. If the number of ECMP paths is k, then the flow number, the output of the function, is within the range of $[0\, k-1]$. Each flow is assigned to a different paths $Hmod(k)$. Figure 3 shows the flow diagram of the algorithm.

Other newer algorithms are LRU-based Caching with Counting (LCC), Fast Switching (FS), Table-based Hashing (TH), Table-based Hashing with Reassignments (THR) and its different versions in bin reassignment, the traffic splitting algorithm based

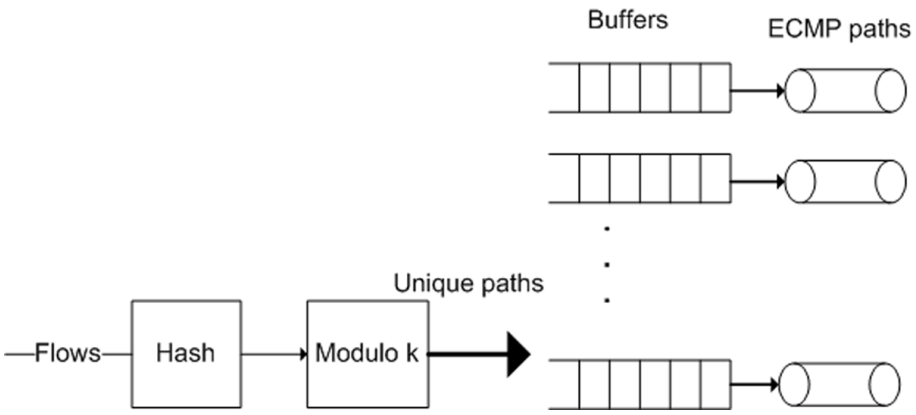

Figure 3: Hash-Threshold Algorithm.

on dual hash table and counters (DHTC), Flowlet Aware Routing Engine (FLARE) and Link-Criticality-based ECMP routing (LCER). Each of these algorithm are described next.

Table-based Hashing is very similar to the Hash-Threshold algorithm. The traffic stream is split into b bins and then b bins are mapped to k paths. A flow diagram of the algorithm is shown in Figure 4. Table-based Hashing has two benefits in contrast to Hash-Threshold. At first, unequal load balancing is easy to implement with Table-based Hashing. Secondly, mappings from bins to paths can be preconfigured. Naturally if $b = k$, the algorithm is the same as Hash-Threshold. [22], [59]

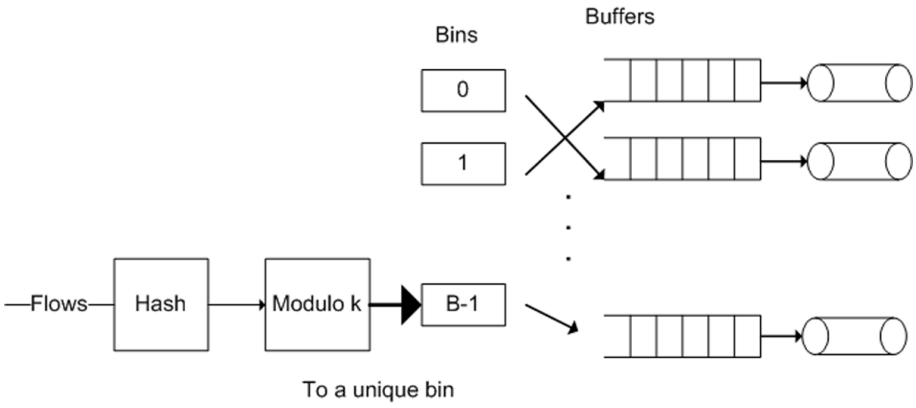

Figure 4: Table-based Hashing.

Table-based Hashing with Reassignment (THR) improves the Table-based Hashing algorithm by dynamically changing mappings based on real-time traffic loads. Traffic loads can be easily measured by assigning counters to each bin. A small amounts of packet disorder is allowed due to the existence of the TCP congestion

window. Bins are reassigned from the heaviest load path to the lightest load path after a pre-defined time interval. Calculation of the heaviest load is based on the moving average. The decision of the time interval is the most vital because it affects on packet ordering and load balancing performance. Using a small time interval, load balancing performs better, but there are more packets out-of-order. THR is shown in Figure 5. [91]

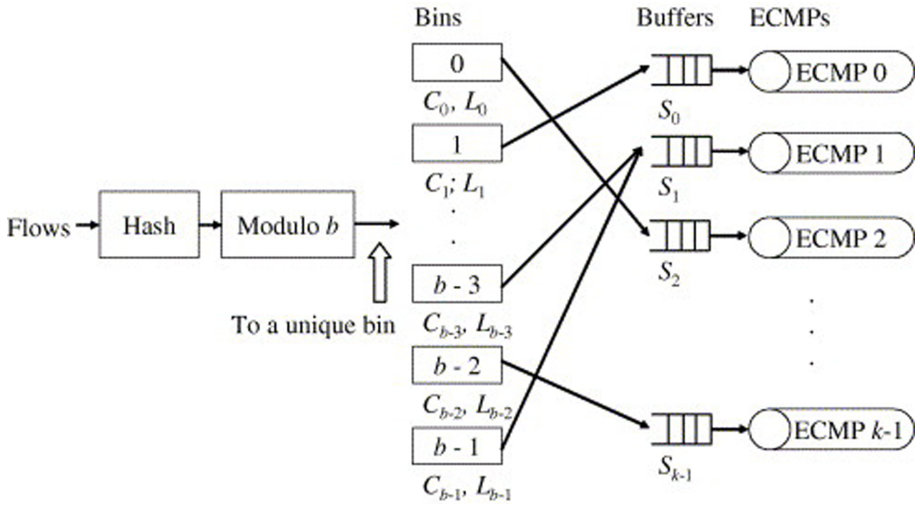

Figure 5: Table-based Hashing with Reassignment.

The authors in [69] and [119] define different bin reassignment algorithms for dynamic load balancing. The basic idea is the same as in the THR algorithm shown in Figure 5. The authors introduce four bin disconnection strategies; Conservative Single Bin Disconnection (SBD^+), Progressive Single Bin Disconnection (SBD^-), Conservative Multiple Bin Disconnection (MBD^+) and Progressive Multiple Bin Disconnection (MBD^-), and two reconnection strategies; Absolute Difference Bin Reconnection (ADBR) and Relative Difference Bin Reconnection (RDBR). All in all, there are eight different algorithms, when each combination of disconnection and reconnection strategies are used. Combination of SBD^- and ADBR is the same as the THR algorithm. The results are that reassignment of multiple bins (MBD) in a single load balancing step outperforms clearly single bin reassignment (SBD). ADBR algorithms are less complex and more accurate than RDBR.

Fast Switching (FS) distributes traffic in a round-robin fashion, but the trick is that flows are stored to a cache and if a new packet belongs to a flow found in the cache, it is assigned to the same path. If flow does not exist in the cache, a new flow is assigned to the next ECMP path in a round-robin fashion. Thus, the algorithm performs in a flow-based fashion, but when the cache is full the performance decreases to the level of a packet-by-packet algorithm and packet disordering increases. [70]

LRU-based Caching with Counting (LCC) is another dynamic algorithm introduced here. Counters similar to THR exist in each ECMP path. LCC treats UDP

and TCP packets differently. UDP packets are directly assigned to the least-loaded ECMP path whereas in TCP packets, flows are kept in the cache and if the flow is new or the time interval of the last packet arrival is so long that the other flow has replaced the old flow, packets are assigned in a least recently used (LRU) manner. LCC is shown in Figures 6 and 7. [59]

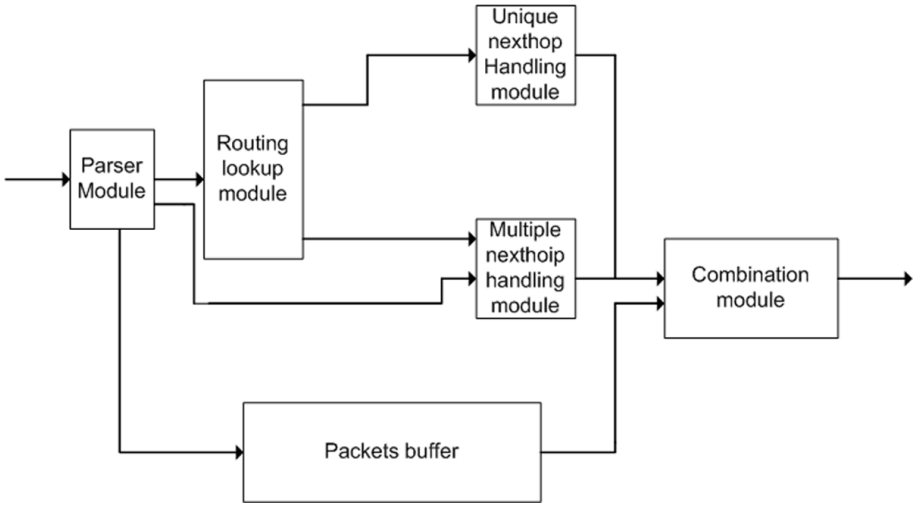

Figure 6: LCC Algorithm.

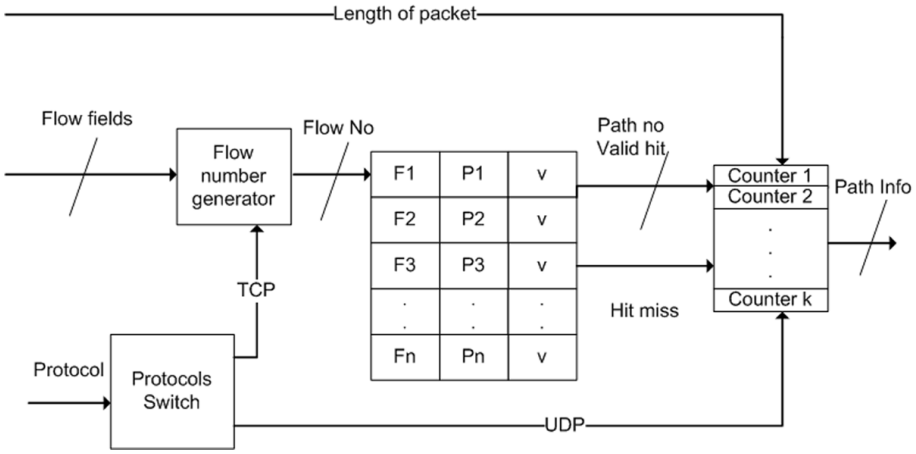

Figure 7: Multiple Nexthop Handling Module of the LCC Algorithm.

Dual hash table and counters (DHTC) is a new traffic splitting algorithm and its research is still unfinished. The first paper has been published by authors in 2010 and it looks promising. The idea of the algorithm is to use counters in each ECMP path and calculate relative values from each path. The new value of the counter is the old value minus the smallest old counter value. The purpose is to take the size of the packet into account. In Ethernet packet size varies from 1518 Bytes to 64 Bytes. The new flow is set to the path having the smallest counter value and

the corresponding counter is updated by the old value plus the packet size. Two hash-tables store all the flows. The primary hash-table stores live flows that are currently sending packets. The secondary hash-table stores flows that collided with the flows in the primary table. Collision means that there is already another flow that has the same path ID in the hash-table. If a collision occurs also in secondary table, the new flow is merged with the existing flow of the secondary table. This has a detrimental effect on traffic distribution, but this is quite a rare case and depends on the size of the hash tables, how often this occurs. This solution makes the best tradeoff between the performance of traffic distribution and the size of memory and complexity of flow mapping. Additionally the variable size of packets is taken into account and there are not packets out-of-order. Results and analysis of this algorithm are still coming up. [68]

Flowlet Aware Routing Engine (FLARE) takes into account that flow can be assigned to any available path without packet reordering, if the time between two successive packets is larger than the maximum delay difference between the parallel paths. Flowlets are packet bursts that are switched in FLARE. The delay difference between the parallel paths is estimated using periodic pings and calculating an exponential running average of the measurements. The basic hash is performed and depending on the measured last_seen_time, packets are assigned to a path of the existing flowlet or to a new path. A token counter exists in each path, that calculates, how far the path is from its desired load. Counters are reset after every 0,5 seconds in order to keep the traffic distribution measurement up-to-date. The token counter compares its measurements to the splitting vector, that may be static, or can be based on any dynamic algorithm.

Link-Criticality-based ECMP routing (LCER) is the last algorithm introduced. The selection of each ECMP path is based on the link's average expected load, link capacity and the path's length. The average expected load is assumed to change on a daily basis and is set from measured daily traffic profiles or service-level agreements. The probability of each path to be selected is based on the following calculation:

$$P_i = a_0 \frac{X_0}{h(i)} + a_1 \frac{d(i)}{X_1} \tag{4}$$

where $a_0 + a_1 = 1, 0 < a_0, a_1 < 1$

$$X_0 = \frac{1}{\sum i = 1^M \frac{1}{h(i)}}, \text{then } \sum i = 1^M \frac{X_0}{h(i)} = 1 \tag{5}$$

$$X_1 = \sum i = 1^M d(i), \ \sum i = 1^M d(i) \frac{d(i)}{X_1} = 1 \tag{6}$$

where the number of ECMP paths is M, X_0 is a constant depending on each ECMP $h_1, h_2, ...h_M$, X_1 is a constant related to the difference of each ECMP, and $d(i)$ is difference the between each ECMP path.

The selection probability of each ECMP path is composed of the hops of ECMP and average utilization of the link. A longer path with higher link average utilization has smaller selection probability. The calculation is done for example once in a day after new measurements, so LCER does not need any online calculation. Actually, LCER is an optimization algorithm that shows the correct probabilities of each ECMP path. The authors do not describe how the implementation part is done, but a static load balancing algorithm, that can change traffic distribution, is suitable for implementation. This kind of algorithm is, for instance, simple table-based hashing algorithm. [92]

All these dynamic algorithms, LCER, LCC, FLARE, THR and its different versions provide better performance than their static counterparts[92],[59],[91], [69]. Small increase of packets out-of-order does not affect much on total performance. LCER is the only algorithm that provides network-wide load balancing performance, no packets arrived out-of-order, and the lowest average end-to-end packet delays [92]. Although, an additional work task is the daily adjustment of traffic splitting vector of every router. The authors in document [98] compare analytically the THR and FLARE algorithms and FLARE's performance in bursty traffic is demonstrated. Because FLARE has the property to mitigate packet reordering almost totally, it outperforms THR. Since FLARE can use any traffic splitting vector, it is a general solution to equal and unequal traffic splitting scenarios. For example, in an ECMP case, traffic in equal cost paths could be split statically based on bandwidth of output interfaces or LCER could provide a more optimal vector on daily basis. In MPLS case, some dynamic algorithms, such as TeXCP, could provide near optimal traffic splitting vector for FLARE.

The authors in [119] studied multi-stage network architecture where load balanced traffic from different origins provides the input for the next load balancer. All hash-type algorithms have one severe problem, that derives from the fact that every router's hash ends up to the same result, when consecutive load balancing occurs in the network. This is called The Traffic Polarization Effect. Figure 8 shows the problem. Router B in the figure performs the same hash as router A and all flows go through the same interface to router D and none of the traffic goes to router E. This can be avoided by adding a random ID, that is specific to each router, to the hash. Another minor phenomenon in a multi-stage network is that dynamic load balancing algorithms reduce the inaccuracy by reassigning flows to other paths and cause thereby another potential for packet reordering. The overall flow reassignment rate increases approximately linearly with the number of load balancing steps. Additionally, the authors propose that load balancing should not be applied too often to the same set of flows since this increases the probability for route flaps and packet reordering.

An extension to sub-optimal single flow TCP is multipath TCP (MPTCP) that decreases the need for load balancing the TCP traffic in the IP layer, and because MPTCP has a much larger receiving buffer than the original TCP, having packets out-of-order is more tolerable. The next section gives a short introduction to MPTCP.

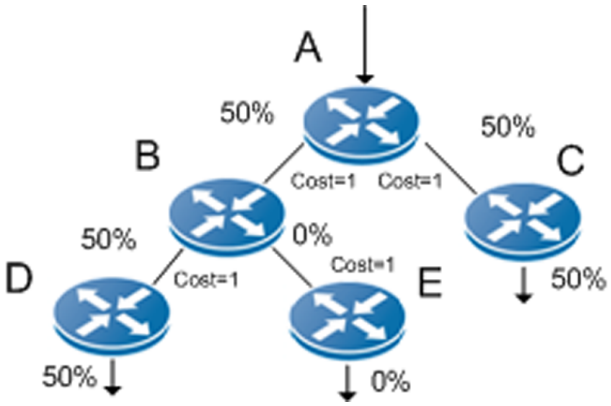

Figure 8: Traffic Polarization Effect of Hash-based Algorithm.

In the section 4.5.5, path calculation algorithms developed for an MPLS network are discussed very briefly. Some of them can be applied also to an IP case, but most of them are so complex, that they do not work as well in hop-by-hop based IP routing.

4.3.2 Multipath TCP

The idea of changing TCP in order to facilitate load balancing design has existed for quite a long time. For example, the authors proposed changes to TCP in document [83] in 2002. Currently, IETF is developing multipath TCP (MPTCP) that enables load balancing in the transport layer. MPTCP splits a TCP connection across multiple paths. The definition of flow changes along with MPTCP. In MPTCP, different flows can be organized in the transport layer. Table 24 shows the new organization of sub-flows. Additional TCP sessions are created for each sub-flow. Multiple paths are identified by the presence of multiple addresses at the endpoints of TCP sessions.

Table 24: MPTCP Protocol Stack

Application	
MPTCP	
Subflow (TCP)	Subflow (TCP)
IP	IP

MPTCP uses a single congestion window at the receiving end. Having a connection level sequence number and a sub-flow level sequence number enables two level packet re-ordering. The receiving buffer must be larger in the MPTCP case than in the single path TCP case. According to [60], the receiver has to buffer all sub-flows for the duration of the highest Round-Trip Time (RTT) or Retransmission TimeOut (RTO) of all subflows in the worst case. As a result, this demand relaxes restrictions of packet re-ordering in the IP layer as well.

When sub-flow fails, data is retransmitted to a different sub-flow. MPTCP provides load balancing tools to control congestion in each sub-flow. Resilience is also improved by transmitting and re-transmitting data on any available path.

Because sending hosts do not have knowledge of existing multiple paths in the IP layer, the use of multiple IP addresses can be possible. Source and destination addresses and probably the port numbers define the path. MPTCP requires also changes to host routing and middleboxes, such as NATs, firewalls, etc. [60], [61], [66]

There are also other transport layer protocols than UDP and TCP. For example, the Stream Control Transmission Protocol (SCTP) ensures reliable transport and congestion control [54]. Similar multipath designs are also available for SCTP. IETF's draft [67] explains more about this topic.

4.4 Other IP protection and load balancing mechanisms

Also other IP load balancing and protection mechanisms exist in addition to ECMP. VRRP, Link Aggregation and IP fast reroute are explained briefly in the following sections.

4.4.1 IP Fast Reroute

ECMP provides general, simple and practical fast protection for IP networks. Nevertheless, ECMP does not cover all the single link and node recovery cases. IP Fast Reroute (IP-FRR) is the IP counterpart for the well-known MPLS fast reroute that is also discussed later in this chapter. Basically, IP-FRR precalculates the alternate next-hop and after detection of a failure of the primary path, traffic is switched quickly to this backup alternate. With the use of IP-FRR, recovery of a few milliseconds is possible. IETF's document [24] explains the use of Loop-Free Alternates (LFA), which increases the scope of failure recovery to about 80%. The primary path is the shortest path and the other paths, that have nexthop closer to destination, are potential backup paths. Only local signaling is needed to discover alternate paths. The solution is loop-free, because after every nexthop, the total cost to destination decreases. It is also possible to use LFAs for load balancing purposes, but this is commonly omitted.

Nevertheless, 100 % failure recovery is not possible with ECMP and LFA only. In order to cover all cases of recovery, more complex techniques needs to be developed. IP-FRR has been studied during the last four years. Several solutions for IP-FRR have been proposed. U-Turn Alternates [62], Tunnels [63], Multiple routing configuration (MRC) [118], [117], [110], efficient scan for alternate paths (ESCAP) [99], failure insensitive routing (FIR) [112] and tunneling using Not-via addresses (Not-via) [115] are the most known techniques. All these approaches except U-Turn alternates and tunnels provide 100% recovery of single link and node failures [107], [108]. Each of these approaches achieves 100% recovery by employing multiple routing tables.

Currently, Not-via's improvement, Lightweight Not-via, [111] is the most promising technique, because it is the only scheme that handles shared link group (SRLG) failures. The Not-Via approach introduces a new address to the router's interface; the Not-Via address, which means that when paths are computed, the path from the router to the Not-Via address is not included in the path. When a failure occurs, router tunnels traffic to Not-Via address and the failure is bypassed. Not-Via needs heavy SPT calculations and lots of management. Lightweight Not-Via extends protection and solves some of the complexity and management problems, but the approach is still quite complex.

4.4.2 Link aggregation

Link aggregation is the IEEE standard defined in IEEE 802.1AX-2008. The purpose of link aggregation is to use multiple links in parallel in order to increase bandwidth and redundancy. Figure 9 shows an example of LAG. The Link Aggregation Control Protocol (LACP) is a control protocol running over Ethernet Link Aggregation Group (LAG) members. LACP automatically detects aggregatable links and bundles them to LAGs. One LAG works as a one higher-capacity link in the upper IP layers. LACP also monitors connectivity and removes malfunctioning links from the LAG. Link monitoring is relatively slow, with fastest detection time being 3 seconds, defined in a variable of Short_Timeout_Time of LACP specification. Actually, this detection helps only against configuration errors, because link faults are already detected in the physical layer. Traffic is distributed to the member links by using a load balancing algorithm. LACP uses a well known multicast group address as destination MAC and link specific source MAC. LACP is specified in IEEE 802.3. Ling aggregation can be used in MPLS over Ethernet, IP over Ethernet and anything over Ethernet. Figure 10 shows these different setups.

Link Aggregation is quite similar to the ECMP feature. Both can use the same load balancing algorithm and both provide protection and load balancing mechanisms. However, LACP is a lower layer protocol (layer 2 in OSI model). LACP provides finer classification of flows than ECMP, because bundling and unbundling of LAG members is performed on the same group of links. ECMP is a more general method of achieving load balancing, since link aggregation cannot support load balancing path that is longer than a single link without using tunneling. Nevertheless, link aggregation is an easy way to increase bandwidth and all unused links in the router can be taken into use.

4.4.3 VRRP

The Virtual Router Redundancy Protocol is an election protocol to increase a node's redundancy. It is specified by IETF in [16]. The idea is to create a virtual router that is actually a cluster of physical routers including a master and one or more backup routers. All packets addressed to the default router's IP addresses and corresponding MAC addresses are sent to the master. Ownership of the virtual IP addresses is

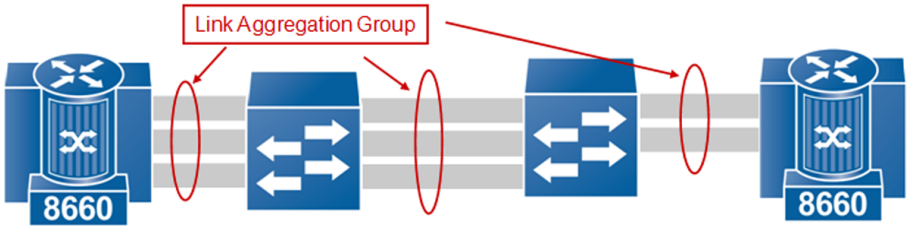

Figure 9: Link Aggregation.

elected based on the priority of the routers. If a master fails, the router with the next highest priority takes ownership of the addresses. VRRP provides higher availability of the default path without additional dynamic routing or router discovery protocols. Additionally, in IPv6, VRRP provides quicker switchover to backup routers than can be obtained with standard IPv6 neighbor discovery mechanisms.

VRRP can enhance load balancing by distributing end nodes across multiple routers. If two VRRP addresses are configured for the interfaces of two VRRP routers and the first router has the highest priority of the first VRRP address and the other router has highest priority of the other address, and if half of the subnet's hosts are set using the first address and the other half are set using the other VRRP address as their default gateway, the subnet's traffic is divided into two different paths and protection works still the same way as in a single VRRP address case. The use of this kind of setup consumes one IP address and one default gateway address per router. It also creates extra configuration with the Dynamic Host Configuration Protocol (DHCP). Cisco's Gateway Load Balancing Protocol (GLBP) solves these problems. When GLBP is enabled, a single default gateway address is enough, but still all hosts are evenly distributed between all VRRP routers.

4.5 MPLS

Multiprotocol Label Switching (MPLS) is a packet switching technology standardized by IETF. Document [131] describes the architecture of MPLS. In contrast to IP packet forwarding, MPLS processing and sorting of packets is performed only in the beginning of a connection. An ingress Label Edge Router (LER) classifies packets based on different information of the IP header, destination port or traffic class, that defines the Forwarding Equivalence Class (FEC) of this traffic flow. Packets in the same FEC are forwarded with the same MPLS label. LER assigns a new MPLS header in front of the original datagram header originated from a conventional network and sends it through the MPLS network, where packets are switched based on the label. Egress LER removes the MPLS label and forwards the packet to conventional destination network. The path through the MPLS network is called the Label Switched Path (LSP). LSP can have explicit paths or paths created by IP routing protocols. Figure 11 shows a typical MPLS network.

Usually an IP packet is encapsulated in the MPLS packet but as the name "mul-

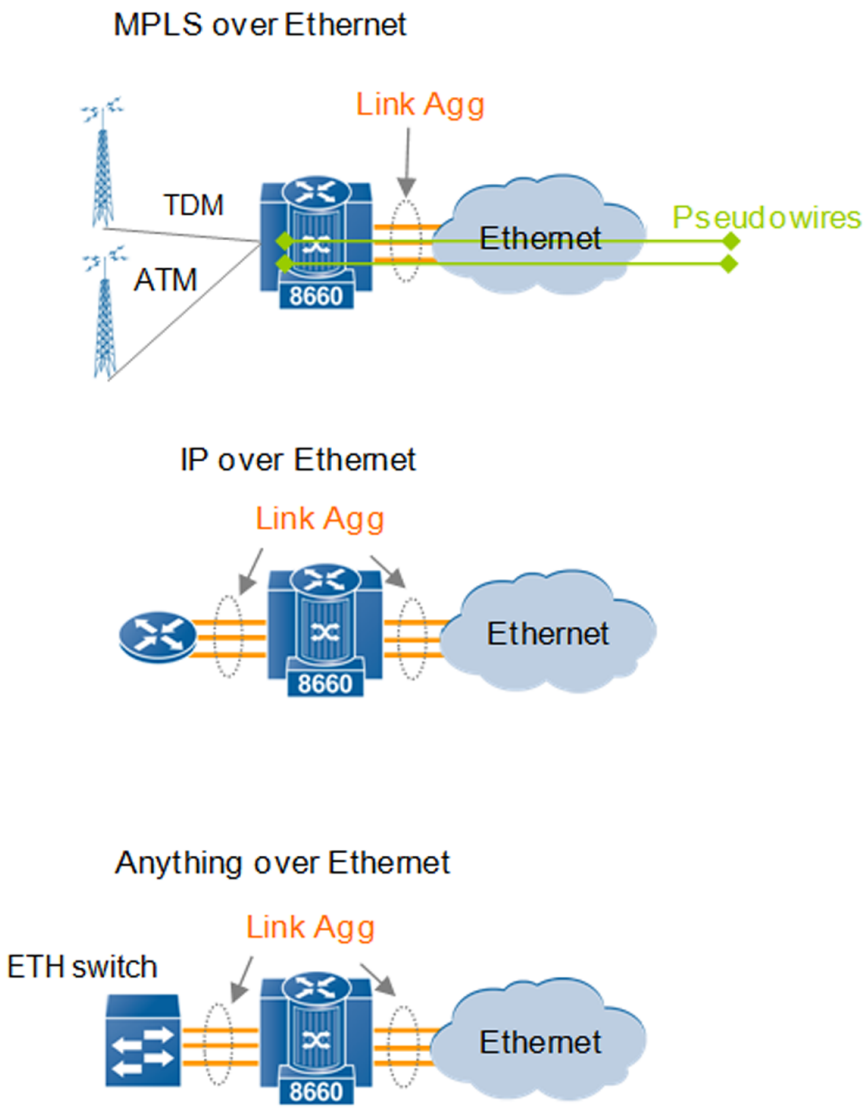

Figure 10: Different Setups of LAG.

tiprotocol" suggests, it can be a frame of any transporting layer and also another MPLS frame, so MPLS can be utilized to create hierarchies. This is one of the most important properties of MPLS and a central part of setting up IP Virtual Private Networks (IP-VPNs). Other important applications of MPLS are BGP-free core networks, layer-2 VPNs such as pseudowires, Virtual Private LAN Service (VPLS) and traffic engineering based on RSVP-TE.

As Figure 12 shows, an MPLS header contains a 20 bit long label, that supports about one million different flows, a 3 bits long traffic class-field (old EXP-field), an S-field, the length of which is one bit, and an 8 bits long Time To Live (TTL) field that has the same purpose as the TTL-field in the IP header. When the S-field is one, it denotes that the current header is the last in the stack. The label and

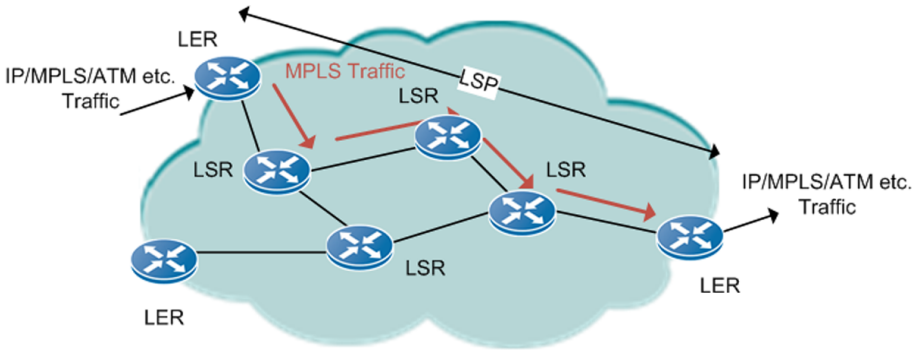

Figure 11: Typical MPLS network.

possible TC-field are enough to provide switching of packets through the MPLS network. The SA and DA of the IP header and forwarding table lookup are not needed anymore and forwarding of packets is simplified. [131], [129], [130], [39]

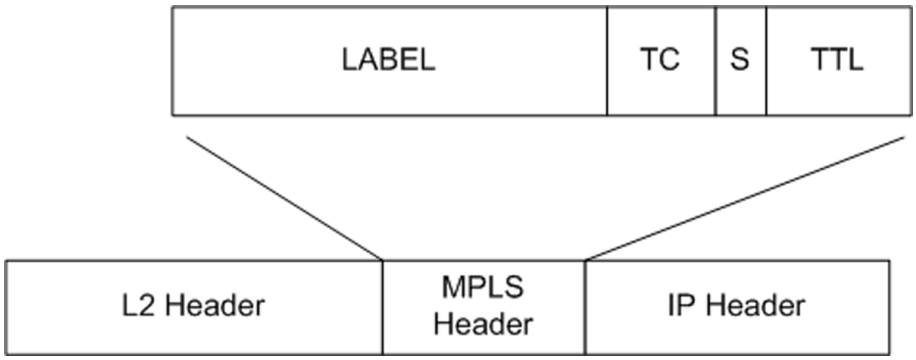

Figure 12: MPLS header.

4.5.1 MPLS signaling

MPLS label values can be established to each LSP using the Label Distribution Protocol (LDP) with an IGP shortest path metric, BGP signaling or using the Resource Reservation Protocol with a traffic engineering extension (RSVP-TE). The Label Distribution Protocol is defined in [25]. It discovers potential peers by using LDP Link Hello messages sent by the User Datagram Protocol (UDP) to find directly connected Label Switching Router (LSR) neighbors or UDP LDP targeted Hello messages to locate LDP peers at specific addresses. LDP establishes sessions between two LSRs using the Transmission Control Protocol (TCP). Notification messages and advertisement messages, that create, change, and delete label mappings for FECs, are distributed to LSRs over the TCP sessions. Distribution of messages can be done either in Downstream on Demand mode, in which an LSR explicitly requests label information from its peer, or Downstream Unsolicited mode, in which label

information is sent without waiting for a request. LDP does not have any routing functionality and it relies totally on information received from the IGP. The LSP path shifts, when the shortest path changes.

From the TE point of view, the most interesting signaling method is RSVP-TE. The Resource Reservation Protocol was designed to create bandwidth and other resource reservations across IP networks. It ensures minimum Quality of Service (QOS) between sending hosts. RSVP-TE signaling can be used for creating explicitly routed point-to-point or point-to-multipoint LSPs. A single LSP requires only one RSVP session containing many end-to-end flows. An ingress LER can specify the entire LSP path or particular nodes which the packets are passing through. The router sends an RSVP path message to the Egress LER, that responds with an Resv message, that uses the exact reverse path of the RSVP path message. The RSVP label message contains the path, the route and the bandwidth reservation requests for the path among others. Periodic message exchange is required to maintain the state of the RSVP session. [42]

In RSVP-TE, non- explicitly routed LSPs are computed using the Constrained Shortest Path First algorithm (CSPF). CSPF uses information from the Traffic Engineering Database (TED) that is an extended version of the link-state database containing the link bandwidth and administrative constraints in addition to the original link-state information. After pruning links, that do not meet the constraints, the shortest path algorithm towards the egress LER is executed. The algorithm may use an IGP metric or link TE metric to determine the shortest path. The TED is built from information flooded by OSPF and IS-IS with a TE extension [26], [27]. These link attribute extensions are shown in Table 25. CSPF does not give optimal paths, but it is considered a reasonable approximation. CSPF computation depends on the accuracy of the TED. Usually the LSP is calculated using CSPF only once. The use of LSP reoptimization when the network state changes is a better option, because traffic is transferred to more a optimal path when such is founded, but it incurs additional instability of the network, which is caused by the shifting of traffic patterns in the network and preemption of LSPs. The third option is to use offline tools, that provide optimal results and can take into account failure cases. The drawbacks of static setting of LSPs are that there is no adaptivity to changes of traffic, accurate information about traffic distribution is needed and upgrade and reconfiguration are problematic. Traffic engineering with MPLS provides different recovery mechanisms that are explained next briefly. [129], [130], [133], [42]

4.5.2 MPLS recovery mechanisms

Fast restoration and protection mechanisms are needed when high QoS demand services are offered. Connection-oriented MPLS LSPs are a candidate for implementing different types of recovery schemes. These path protection techniques between the ingress and the egress edge LSRs can be accommodated by signaling two explicitly routed point-to-point LSPs, traversing diverse physical routes. Since the traffic rerouting decisions and protection LSP setup procedures have been performed al-

Table 25: Traffic Engineering Extensions to IS-IS and OSPF

IS-IS Sub-TLV	**Type Number**	**Length (Bytes)**	**Corresponding OSPF Sub-TLV**
Administrative Group (Color)	3	4	Administrative Group
IPv4 Interface Address	6	4	Local Interface IP Address
IPv4 Neighbor Address	8	4	Remote Interface IP Address
Maximum Link Bandwidth	9	4	Maximum Bandwidth
Reservable Link Bandwidth	10	4	Maximum Reservable Bandwidth
Unreserved Bandwidth	11	32	Unreserved Bandwidth
TE Default Metric	18	3	TE Metric

ready before any failure occurs, these techniques enable significantly faster network and traffic restoration times than normal traffic rerouting functionality.

This section provides a definition of different protection and restoration types. Using IETF's terms, recovery means both restoration and protection. The distinction between restoration and protection is that in protection, there is no need for signaling when a failure occurs, because route computation, resource allocation, cross-connection and backup LSPs have been established beforehand. However, other signaling such as fault notification and synchronization is needed during the event of failure. In restoration, additional signaling is needed to establish a recovery path. The recovery and protection can be managed locally, on a segment basis or on an end-to-end-basis. End-to-end protection means protection of the whole LSP, a segment refers to the portion of the LSP and local protection refers to the link or span between the two nodes. The terminology, and also protection and restoration mechanisms between these three are very similar, although the span recovery does not protect the nodes at the link's both ends. Different recovery types are available:

- $1 + 1$ type: dedicated protection

  This is only a protection type. Normal traffic is duplicated to two different spans/LSPs.

- $0 : 1$ type: unprotected

  This is only a restoration mechanism. There is no specific pre-computed route nor pre-established recourses. LSPs are transferred dynamically to another span.

- $1 : 1$ type: dedicated recovery with extra traffic

A specific backup path protects one working span/LSP. There is no traffic duplication. This type of mechanism works in LSP/span protection and LSP restoration.

- $1 : N (N > 1)$ type: shared recovery with extra traffic

  N specific working LSPs/spans are protected by one specific LSP/span. Recovery span/LSP can transport extra traffic. Only one LSP/span of the N LSPs/spans can be recovered. This type of mechanism works in LSP/span protection and LSP restoration. Every LSP/span must have the same ingress LSR and the same egress LSR.

- $M : N (M, N > 1, N \geq M)$ type

  N specific working LSPs/spans are protected by M specific recovery LSPs/spans. Recovery spans/LSPs can transport extra traffic. M LSPs/spans of the N LSPs/spans can be recovered. This type of mechanism works in LSP/span protection and LSP restoration. Every LSP/span must have the same ingress LSR and the same egress LSR.

Recovery paths can be unidirectional or bi-directional. In general, 1+1 type protection schemes are usually the fastest, but also they use the most capacity of the network and protections that use less capacity are not as fast as 1+1 type to recover from failure.

According to [28], there are four phases of recovery;

- Failure detection
- Failure localization and isolation
- Failure notification
- Recovery from failure

Failure detection and Bidirectional Forwarding Detection (BFD) are described in section 4.7. Failure localization is needed to identify the scale of failure, to decide whether local recovery is enough and decide how far the notification must be sent. More detailed information of recovery mechanisms in (G)MPLS networks is available, for example, in IETF's documents [28], [29], [30] and [31]. The next section explains a fast local recovery mechanism called MPLS fast reroute.

4.5.3 MPLS Fast Reroute

In an MPLS network, an RSVP-signaled LSP can be protected from single node and link failure using MPLS Fast Reroute (MPLS-FRR). The idea behind MPLS-FRR is similar to IP-FRR: Signaling the backup path beforehand and switching the traffic to it, when failure of primary path occurs. Local repair guarantees a fast failure

response and recovery close to the point of failure. A single pre-established LSP can protect one or multiple LSPs.

Document [33] defines two different modes of MPLS-FRR, one-to-one backup and facility backup. In facility backup, a single LSP called bypass tunnel is created to back up a set of LSPs. This is possible, if a bypass tunnel intersects the path of the protected LSP downstream and protected LSPs pass through the Point of Local Repair (PLR) and common downstream node. A PLR is a node that redirects the traffic from the primary path to the preset backup path. In one-to-one backup mode, a separate backup LSP, known as Detour, is established on each LSP at each PLR. Each node along the protected path is a potential PLRs, therefore Detour LSPs must be signaled on each of these nodes. Facility backup is slightly easier to maintain and it is more commonly used but overall, these two modes are very similar.

4.5.4 Multipath treatment in MPLS

The multipath-idea has also been brought to MPLS networks. Nowadays LSPs contain heavy traffic trunks and multipath load balancing helps distribute these heavy traffic loads across the whole network. In MPLS, flow definition is based on FEC. Load balancing can be encoded in two different ways in a label stack. Multiple labels can be created for a particular FEC, or load balancing information can be encoded in a separate label in the label stack. Traffic is split between different LSPs. Flows can be split into multiple sub-flows in order to improve load balancing further. Additionally, multiple paths can be backup paths for one primary path. Two basic cases of setting MPLS multipath are searching for the IP header heurestically from the MPLS label and finding ECMP paths in LDP signaling or setting multiple LSPs to the same destination using RSVP-TE and the user defined traffic splitting of each path.

Document [65] explains the use of multipath LSPs (MPLSP) with RSVP-TE signaling. RSVP-TE and constraint based routing enables exploiting path's resources better. There are several options how ECMP TE LSPs can be created in addition to two basic cases mentioned. The ingress LER can compute all paths of sub-LSPs, LSRs can compute the paths to further downstream ([38]), an RSVP path message can contain path information of one or more paths of the LSP or sub-LSP, or multipath LSPs can have equal cost paths from ingress to egress. Normally traffic is split in proportion to the paths' minimum bandwidth between LSPs. Since the ingress LER has more knowledge of the traffic class, about the network and other constraints, it decides how many MPLSPs are created, which path they take and so on.

The use of multipath in transport networks with all supported Operations, Administration, and Maintenance (OAM) functions is the most demanding. Especially connectivity check (CC), connectivity verification (CV), loss measurement (LM) and delay measurement (DM) cause restrictions on multipath decisions. Document [64]

explains these issues.

Link bundling is defined in [34]. It is an MPLS specific technique for load balancing. It also offers vendor interoperability. In link bundling, multiple point-to-point links are collected together into one logical link. An LSP can be used in one component link of the link bundle, or the LSP can contain all components in the bundle, in which case load balancing is performed between different component links.

One common method in MPLS load balancing is the use of entropy labels defined in [116]. Basically the idea is to add one label to the label stack in the ingress LER that has better knowledge of FEC and IP level forwarding. The label does not have any other purpose than increase the entropy of labels in order to achieve better load balancing.

The next section introduces MPLS path calculation algorithms.

4.5.5 Introduction to MPLS path calculation algorithms

In recent years, quite a few path calculation algorithms have been published and usually the aspired objective is an MPLS network. These normally quite complex solutions provide good load balancing performance. In an MPLS network, flow definition and probably also traffic splitting is performed in the ingress LER, thus more complex solutions can be tolerated and also network convergence is faster than in the IGP case, where load balancing is performed on a hop-by-hop basis.

A single constraint MPLS routing problem can be solved using arbitrary traffic splitting and linear programming, but commonly the number of LSPs and splitting ratio are much too high to utilize it in practise. As already mentioned, a basic approach for setting up LSPs is based on constraint-based routing (CBR) and individual traffic trunks. Other routing schemes are, for example, Widest Shortest Path (WSP) [96] and Shortest Widest Path (SWP)[95].

In QOS TE, optimization of LSP paths using multiple constraints, such as delay, jitter or packet loss, is NP hard [109]. Several algorithms provide QOS optimization. Additionally, point-to-multipoint and multipoint-to-point algorithms exist. These areas are omitted from more detailed discussion.

Again, MPLS path calculation algorithms can be divided into online and offline TE and the same challenges and benefits exist as mentioned in the earlier IP TE sections. A new feature in the MPLS case is the re-routing performance, which means that the impact on LSPs in other parts of the network should be kept minimal when a link or a node failure occurs.

MPLS path calculation algorithms can be divided into three different categories, oblivious routing algorithms, minimum interference algorithms and prediction-based algorithms. In oblivious routing, optimization is based on the worst case performance of a set of traffic matrixes. Path selections of oblivious algorithms are independent of the current traffic load and thus they have the potential to handle traffic spikes well. The drawback of this approach is that those kinds of algorithms

do not perform as well in normal traffic situations as prediction-based algorithms. Furthermore, the authors in [104] observed that path dispersion and path variation are generally high in oblivious routing.

In prediction based algorithms, that try to achieve near optimal routing, the traffic matrix of the network has to be measured or predicted. With the help of TM, optimal routing algorithms solve the problem successfully in stable and small networks. In cases where changes of traffic are rapid, measurement and prediction of TM become difficult. Additionally, data collection to TM becomes challenging in large networks.

Online algorithms, such as MATE [105], S-MATE [72] and TeXCP [100], are the extreme case of prediction-based algorithms. They react to realtime traffic demands and failures. These algorithms converge quickly and they do not need to collect many samples. Load balancing performance is good, but they suffer from large transient penalties in significant and fast traffic changes.

Some of the algorithms try to combine oblivious routing and online routing. COPE [101] is an example of this kind of algorithm.

Minimum interference algorithms are based on a link's criticality, the same principle on which LCER is based on. LCER is described in section 4.3.1. Algorithms try to avoid links that are most likely used in other LSPs. MIRA [102], BU-MIRA [103] and DORA [106] are examples of these kinds of algorithms.

4.5.6 Overlay routing

Overlay routing is an alternative approach to TE load balancing. In overlay routing, service providers establish logical connections between the edge nodes and form a full-mesh virtual network on top of the physical topology. Traffic distribution in the network can be controlled then by routing these logical connections. Now it is possible to tune the performance and availability of any observed path without relying on the underlying routing infrastructure. Using a linear programming formulation, it is possible to achieve optimal mapping between logical connections and physical links. Nevertheless, there are severe problems with overlay routing. First, it does not scale well to large networks. If the number of routers is N, then the number of LSP paths that have to be set up in a fully-meshed network is in the order of N^2. Managing LSPs becomes an exhausting task.

Hybrid or integrated models of overlay and TE load balancing have been developed to overcome the scalability issues of overlay routing.

4.6 IP-TE and MPLS-TE comparison

Surprisingly, traffic engineering was first introduced in MPLS-based environments [19]. MPLS-TE has the advantage of explicit routing and arbitrary splitting of traffic, which is highly flexible for both routing and forwarding optimization purposes.

In explicit routing, traffic is delivered down a path that can be defined entirely in advance. Additionally, certain links or nodes can be defined not to be included in any LSPs. It can be argued, that by setting the link metric to a very high value, the link or node could be bypassed also in an IP-TE environment. Management, scalability and robustness of LSPs are the drawbacks of MPLS-TE. The number of LSPs grows easily too large to manage and creating backup paths for robustness increases the amount of LSPs even more. If full-mesh LSPs are set up, IETF's document [35] lightens the burden of LSP management by providing automatic discovery of the set of LSR members of a mesh.

IP-TE is a highly robust technology. In a very unstable environment, IGPs find always the path to destination, even if only one path exists. Configuring and management of IGPs is easy and simple. Introducing ECMP does not increase any additional configuration. When optimal weights are utilized to IGP, weights have to be set to the whole network using a management tool, which is the most time-consuming operation in IP-TE.

From the point of view of network design, MPLS-TE is easier, because the changing of metrics has a network-wide effect in IP-TE, whereas explicitly routed LSPs can be set up independently.

Nowadays both IP-FRR and MPLS-FRR support fast protection and MPLSP and ECMP provide load balancing for both environments and thus, decision between these different technologies should be based on other factors. IP-TE provides a simpler solution, and if QoS, VPN technologies or other services provided by MPLS are not needed, IP-TE provides a more cost-efficient solution.

4.7 Failure detection: BFD

Bidirectional Forwarding Detection (BFD) is one of the key protocols enabling fast protection and recovery from failures. It is specified in [36]. Basically, BFD is a simple hello protocol designed for fast failure detection. BFD can be used for any kind of links, LSPs or tunnels between forwarding systems. Explicit configuration of end systems is needed, because BFD does not have any discovery mechanism. It has two operating modes, asynchronous mode and demand mode. In asynchronous mode, packets are transmitted at periodic intervals to both directions and after a defined number of lost packets, a path is assumed to be failed. In demand mode, connectivity is verified only when either of the systems needs to explicitly verify it. This system starts a poll sequence, ie. by sending periodically control packets, and waits until a response is received from the other system, or until detection time expires, and the path is declared to be down.

An echo function improves the performance of both modes of operation. Now the system sends a stream of packets to other systems that then sends them back. If a defined number of packets are lost in the stream, the path is declared to be down. The transmitting rate of the stream is negotiated before the actual streaming.

5 ECMP extensions to existing architecture

This chapter explains changes to parts of the software and hardware that are affected by the ECMP feature and are currently supporting single-path routing only. The most important new hardware component is the load balancing algorithm and the software components worth mentioning are SPF calculations for finding alternative routes in IS-IS and OSPF, extensions to manipulate the routing table and the forwarding table and a fast mechanism of indicating changes of paths from software to the load balancing algorithm. Additionally, any parts of the software assuming a single-hop or single path should be naturally changed.

Different kinds of load balancing algorithms are explained in section 4.3.2. Changes to the SPF calculation are explained in section 3.1. As mentioned, the complexity of SPF does not increase much. In truth, the only issue to reckon with is the growth of the routing and forwarding tables and other related data structures. The memory consumption could increase considerably, if the network topology and metrics are suitable for ECMP. Memory optimization of ECMP structures should be designed carefully in order to mitigate the problem.

The author's part of the implementation project was writing a code related to logical ECMP group management. This includes advertisement of multiple nexthops from OSPF and IS-IS routing protocol processes to a general control process, manual static route configuration to the RIB, updates to route selection mechanisms of the RIB and the FIB processes and advertisement of the correct number of ECMP routes to lower abstraction layers of software and to the other control card for redundancy. The RIB should contain all routes and every route should contain all ECMP nexthops configured. There is no limitations of static ECMP routes in the RIB. Nevertheless, only eight of them that have the same destination and distance, are installed to the FIB. The FIB process was needed to change so that the lower abstraction layers of software receive the correct number of existing nexthops. Additionally, new CLI commands were added. Maximum-paths- command configures the maximum number of nexthops advertised from that particular instance of the routing protocol process to the control process.

Probably the most time consuming part was changing the logic of nexthop handling in the control process. The control process has to update the state of all nexthops in a route and additional data structures are needed, because a route is not tied to a single interface anymore.

Naturally, not all static routes that have the same distance and destination are treated as ECMP group. Example of that kind of exception is a recursive route that performs several recursive lookups from the RIB before solving the correct nexthop for the route.

The high level description of the ECMP architecture is described in Figure 13. Lower abstraction layers need also changes due to several nexthops in a single route. Data structures and route processing needs to be changed. Load balancing table controlling the hardware's load balancing algorithm needs to be created etc. Never-

theless, these lower layers are more architecture specific and thus not so important in the context of the thesis.

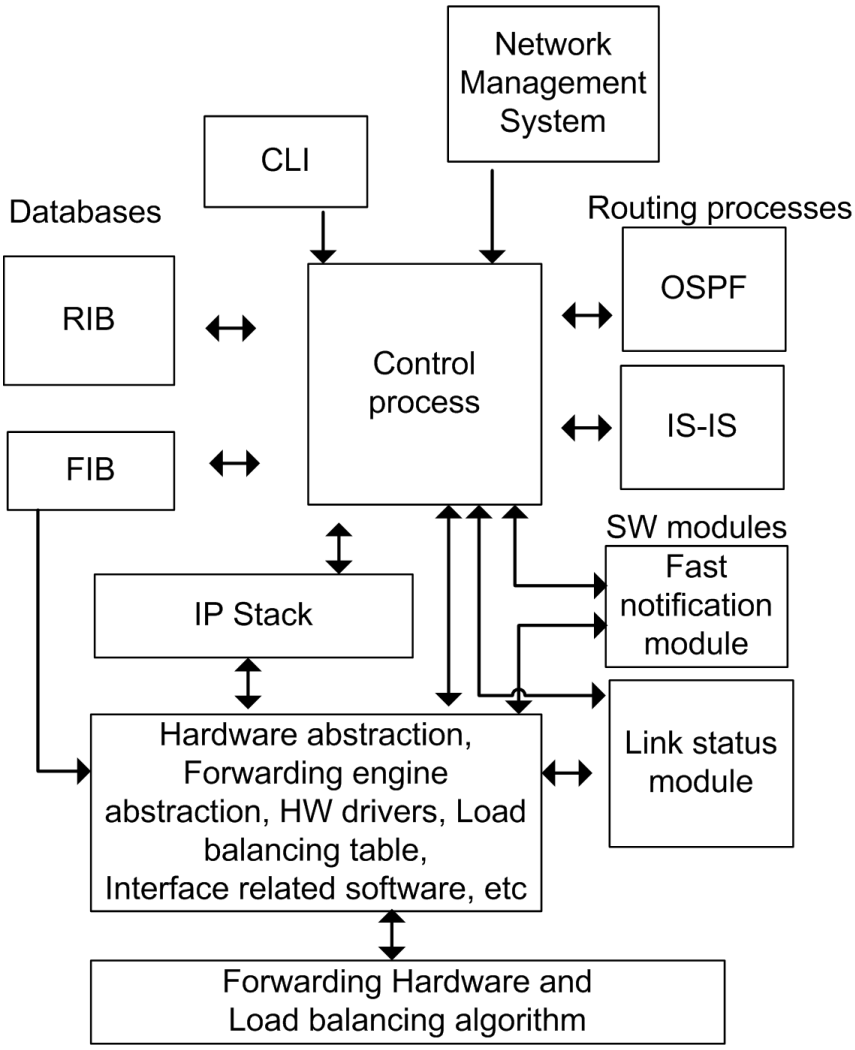

Figure 13: High level description of router's architecture from ECMP point of view.

6 ECMP Configuration and testing

This chapter provides information about the test equipment and configurations used in testing. The purpose of the tests was to investigate the load balancing and fast protection features of newly implemented ECMP software and hardware of Tellabs routers. The performance of the load balancing algorithm was one of the key points. In the load balancing part, traffic distribution between the different links and the value of the bandwidth increase in comparison to a single path case was tested. In the fast protection part, the value of packet loss was measured, when one of the ECMP links was removed.

The size of the network was kept to the minimum in order to keep everything under control.

6.1 ECMP test equipment

The test equipment is shown in Figure 14. It consists of two Tellabs routers and an Agilent N2x tester. The routers are Tellabs 8620 access switch and Tellabs 8630 access switch. There are nine ECMP links between the routers. Load balancing and fast protection features should work between these links if the cost is configured to be equal in all ECMP links. The tester is connected to both routers in order to create transmitting and receiving links. Links from the tester to both routers are operating with bandwidth of 100Mb/s, and ECMP links between the routers are set to operate with a bandwidth of 10Mb/s. Currently, the maximum supported ECMP paths are eight. Thus, the maximum theoretical bandwidth of all links between routers is 80Mb/s. Five ECMP links are installed to the first Fast Ethernet (FE) module and four links to another FE module. The module setup is the same for the 8630 router. Both modules are installed in the same line card.

In fast protection tests, a switch is added between the routers in order to test failure detection using BFD, OSPF and IS-IS in conjuction with ECMP.

6.2 Configuration and testing of static routes

The following command line interface (CLI) commands are used for the creation of nine static routes:

ip route 192.4.1.2/16 6.6.1.2
ip route 192.4.1.2/16 6.6.2.2
ip route 192.4.1.2/16 6.6.3.2
ip route 192.4.1.2/16 6.6.4.2
ip route 192.4.1.2/16 6.6.5.2
ip route 192.4.1.2/16 6.6.6.2
ip route 192.4.1.2/16 6.6.7.2
ip route 192.4.1.2/16 6.6.8.2

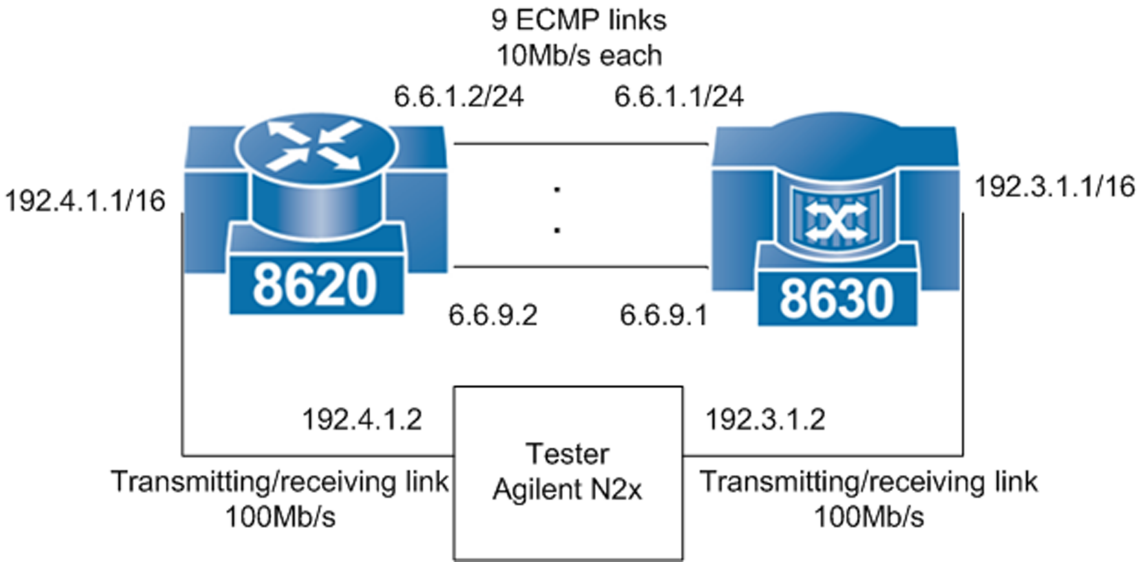

Figure 14: Test Setup.

ip route 192.4.1.2/16 6.6.9.2

The first IP address is the prefix of the destination address and the second one is the nexthop address. The distance of the route is one by default. The general IP route command syntax of Tellabs routers is the following:

```
ip route command syntax:
[no] ip route A.B.C.D/M interface [ distance ]
[no] ip route A.B.C.D/M A.B.C.D [dst-vrf {dest-vrf | __global__ }]
[no] ip route A.B.C.D/M gateway-ip [dst-vrf {dest-vrf | __global__ }]
[ recursive | recursive-mpls ] [ distance ]
```

where the notations are shown in the command description table A1 and convention description table A2 in appendix part of the thesis.

Now we can see the installed ECMP routes using command "show ip route ":

ala-8620-1(config)#show ip route
Codes: C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area, D - OSPF discard
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default

C 6.6.1.0/24 is directly connected, fe0/0, 00:15:33
C 6.6.2.0/24 is directly connected, fe0/1, 00:15:33

C 6.6.3.0/24 is directly connected, fe0/2, 00:15:32
C 6.6.4.0/24 is directly connected, fe0/3, 00:15:32
C 6.6.5.0/24 is directly connected, fe1/4, 00:15:32
C 6.6.6.0/24 is directly connected, fe1/5, 00:15:31
C 6.6.7.0/24 is directly connected, fe1/6, 00:15:31
C 6.6.8.0/24 is directly connected, fe1/7, 00:15:30
C 6.6.9.0/24 is directly connected, fe0/7, 00:15:32
C 10.146.99.74/32 is directly connected, lo0, 00:15:36
C 192.3.1.0/16 is directly connected, fe1/2, 00:15:32
S 192.4.1.0/16 [1\0] via 6.6.1.1, fe0/0, 00:03:04
[1\0] via 6.6.2.1, fe0/1, 00:03:07
[1\0] via 6.6.3.1, fe0/2, 00:03:10
[1\0] via 6.6.4.1, fe0/3, 00:03:14
[1\0] via 6.6.5.1, fe1/4, 00:03:17
[1\0] via 6.6.6.1, fe1/5, 00:03:21
[1\0] via 6.6.7.1, fe1/6, 00:03:24
[1\0] via 6.6.8.1, fe1/7, 00:03:27

6.2.1 Static configuration with BFD

In BFD configuration, each interface needs to be configured and if routes need to be monitored, the bfd command needs to be added in the route configuration:

ip route 192.4.1.0/16 6.6.1.1 bfd
ip route 192.4.1.0/16 6.6.2.1 bfd
ip route 192.4.1.0/16 6.6.3.1 bfd
ip route 192.4.1.0/16 6.6.4.1 bfd
ip route 192.4.1.0/16 6.6.5.1 bfd
ip route 192.4.1.0/16 6.6.6.1 bfd
ip route 192.4.1.0/16 6.6.7.1 bfd
ip route 192.4.1.0/16 6.6.8.1 bfd
ip route 192.4.1.0/16 6.6.9.1 bfd

All interfaces are configured in the following way:

interface fe1/7
no shutdown
ip address 6.6.8.2/24
ip bfd 6.6.8.1 50 3
mode speed 10 duplex full
bandwidth 10M

The BFD interval is 50ms and multiplier is 3. Naturally, the opposite interface of the link also needs to be configured similarly. Checking the bfd state can be done using the command "show ip bfd ":

show ip bfd
fe0/0 6.6.1.1 state UP
fe0/1 6.6.2.1 state UP
fe0/2 6.6.3.1 state UP
fe0/3 6.6.4.1 state UP
fe0/7 6.6.9.1 state UP
fe1/4 6.6.5.1 state UP
fe1/5 6.6.6.1 state UP
fe1/6 6.6.7.1 state UP
fe1/7 6.6.8.1 state UP

6.3 OSPF configuration with ECMP

The basic OSPF settings are the following:

router ospf 1
timers spf 100 100 3000
timers pacing flood 5
timers lsa refresh 0 100 2000
hello-reply 30
network 6.6.0.0/16 area 0.0.0.0
network 10.146.99.169/32 area 0.0.0.0
network 192.4.1.0/24 area 0.0.0.0

Normally, all settings of timers are not mandatory. These settings of timers are used in order to speed up flooding and SPF calculation although they do not have much effect on convergence time in as minimal a network as this. Intelligent timers take the parameters in the order of init, mul and max. Every device is connected to the backbone. There is no need to create different areas. The setting "Hello-reply 30 "enables immediately replying hello that is defined in [43]. A maximum of 30 hello replies per hello interval for all interfaces is permitted. Interfaces are configured in the following way:
interface fe0/0
bandwidth 10M
no shutdown
ip address 6.6.1.2/24
ip ospf hello-interval 1
ip ospf dead-interval 3
mode speed 10 duplex full

The hello-interval is set to 1 second that is the minimum value of the protocol, and the dead-interval is set to 3 seconds. One line in the OSPF router configuration is enough to enable BFD for OSPF:

```
router ospf 1
bfd 50 3
```

This enables BFD per OSPF process. BFD can be enabled also per interface.

6.4 IS-IS configuration with ECMP

The basic configuration of IS-IS is the following:

```
router isis
lsp-gen-interval 3000 100 100
spf-interval 2000 100 50
net 49.0001.1720.1910.2074.00
```

The LSP timer and SPF timer are set a little faster from their default values in order to speed up flooding. IS-IS intelligent timers take the parameters in the order of max, init and mul. In contrast to OSPF, The LSP generation time applies also to the first creation of any LSP. LSPs are generally larger than LSAs, therefore timers should be set to generate advertisements slightly less often than in OSPF. IS-IS needs to be enabled in each interface:

```
interface fe0/0
bandwidth 10M
no shutdown
ip address 6.6.1.2/24
ip router isis
mode speed 10 duplex full
```

The same single line addition to the router's configuration enables process based BFD for IS-IS:

```
router isis
bfd 50 2
```

6.5 Load Balancing Testing

In the testing of the load balancing feature, traffic distribution between different links was measured. Furthermore, the limits of the maximum bandwidth without packet loss with a different number of flows and different number of ECMP links was tested. In these basic tests, all the used flows were of a constant bit rate and the transmitted packets were equal size UDP packets. Different flows were created altering the source addresses of IP packets. For example, two flows means two different source addresses of the whole constant bit rate traffic.

Also a couple of additional tests using bursty traffic were performed. In the first test, the length of the single shot burst was 300 frames and burst load was 100%, which means the burst's inter-departure time (IDT) of 95480974ns and frame IDT of 8160ns. The number of flows was 10.

In the last load balancing tests, variable traffic streams were tested. There were three different streams, the first containing 10 flows with the load of a 8Mb/s, the second stream containing 4 heavy flows with a 20Mb/s load and the last stream containing 20 flows with a load of 5Mb/s. In another test, a single burst with a single flow of average of 9Mb/s burst load (burst IDT=707298524ns and frame IDT=907667) was sent to a link that had also three different streams defined in Table 26.

Table 26: Stream Group Tests

Stream group test 1			The number of ECMP links needed to achieve zero packet loss
stream	load	number of flows	
stream 1	8Mb/s	10	
stream 2	20Mb/s	4	
stream 3	5Mb/s	20	7 links
Stream group test 2			
stream 1	9Mb/s burst	1	
stream 2	5Mb/s	100	
stream 3	20Mb/s	4	
stream 4	10Mb/s	20	8 links

6.6 Fast Protection Testing

The fast protection characteristics of ECMP were tested using static routes with physical layer detection, static routes with BFD and dynamic routes created by OSPF and IS-IS with and without BFD. The traffic was constant bit rate UDP packets with 100 different flows. A single link cable between the routers was removed and the packet loss was measured. Recovery time was then easy to calculate from the knowledge of the packet loss and constant bit rate traffic. In BFD and IGP tests, the switch was between the routers in order to prevent failure detection in the

physical layer. When a link fails, BFD or IGP detects the failure and the traffic should transfer quickly to the rest of the equal cost paths.

Because the traffic is constant bit rate and it is distributed between the ECMP links, the actual bit rate of each link is the original bit rate divided by the proportional traffic distribution of that link, which was measured in the load balancing tests. Thus, following equation gives the right results for recovery time:

$$T_{rec} = \frac{L_p}{B * D} \tag{7}$$

, where T_{rec} is recovery time, L_p is number of packets lost, D is distribution of traffic in the measured link that was measured in the load balancing tests, and B is the original constant bit rate in units of packets per second. For example, if the speed of the traffic is 1000 packets/s, the measured packet loss is 9 and the number of ECMP links is 3, which means a distribution of 39% shown in Table 27, the recovery time is:
$T_{rec} = \frac{9\text{packets}}{1000\text{packets/s}*0,39} = 0.023s = 23ms$

Table 29 is built this way. Average of all results from different number of ECMP links were calculated.

7 Results and analysis

7.1 Load balancing results and analysis

The results of the first load balancing test are in Figure 15. Load balancing performs better, when the number of flows increases. Moreover the available bandwidth grows, when the number of ECMP links increases. When eight paths and 500 flows are used, load utilization reaches almost to the theoretical maximum value of the available bandwidth that is 80Mb/s in this case. The measured value was 79Mb/s. Paths benefit from load balancing immediately, when the number of flows is more than one.

The second test, that is in Table 27, observes load distribution between the ECMP links. It shows that traffic is not always equally distributed. The best results are measured, when the number of links is two, four or eight. The explanation of the behavior is that the load balancing algorithm tries to distribute the traffic to eight different hash results by default. If that many paths do not exist, algorithm tries to fit eight original paths to the existing number of paths. In percent, each of the eight paths corresponds 12.5% of the total load. If, for example six ECMP paths exist, each path will receive 12.5% of the total traffic. Then two paths will receive additional 12.5% traffic load and eventual traffic distribution is shown in the six ECMP calculation line of Table 27.

Table 27: Traffic Distribution with Different Number of ECMP Links

Paths	1	2	3	4	5	6	7	8
2 ECMP Meas	50,0%	50,0%						
2 ECMP Calc	50,0%	50,0%						
3 ECMP Meas	27,0%	34,0%	39,0%					
3 ECMP Calc	25,0%	37,5%	37,5%					
4 ECMP Meas	27,0%	23,0%	23,0%	27,0%				
4 ECMP Calc	25,0%	25,0%	25,0%	25,0%				
5 ECMP Meas	13,0%	14,0%	23,0%	23,0%	27,0%			
5 ECMP Calc	12,5%	12,5%	25,0%	25,0%	25,0%			
6 ECMP Meas	12,0%	11,0%	13,0%	14,0%	27,0%	23,0%		
6 ECMP Calc	12,5%	12,5%	12,5%	12,5%	25,0%	25,0%		
7 ECMP Meas	11,0%	12,0%	14,0%	13,0%	11,0%	12,0%	27,0%	
7 ECMP Calc	12,5%	12,5%	12,5%	12,5%	12,5%	12,5%	25,0%	
8 ECMP Meas	13,0%	14,0%	12,0%	11,0%	11,0%	12,0%	14,0%	13,0%
8 ECMP Calc	12,5%	12,5%	12,5%	12,5%	12,5%	12,5%	12,5%	12,5%

Now if we return to investigate Figure 15, and observe the effect of the number of ECMP links, two, four and eight links provide the best results. The increase of the maximum bandwidth is much larger than in case of other number of links. The

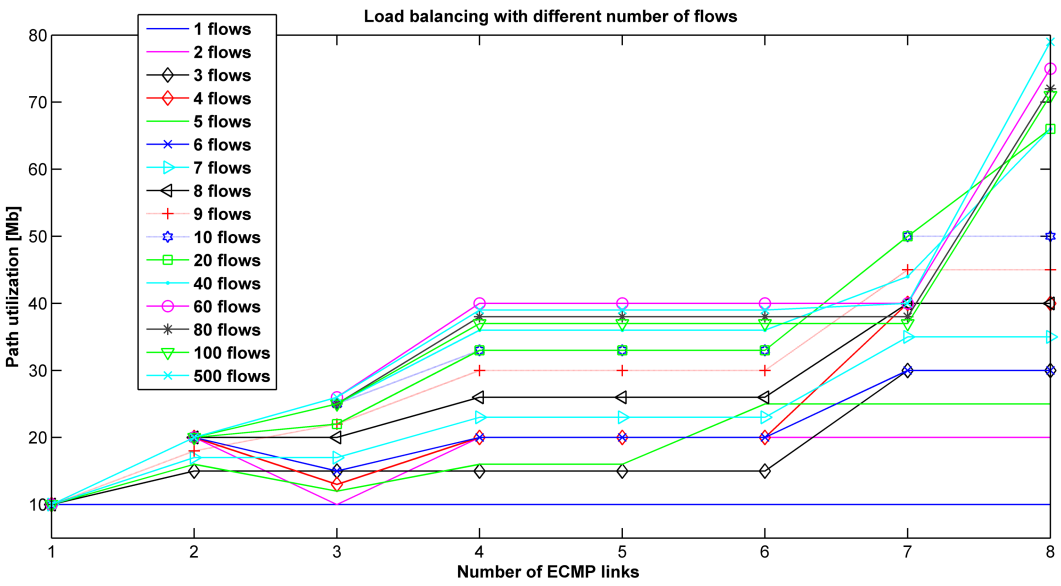

Figure 15: The effect of different number of flows on total utilization of ECMP links.

reason for this behavior is unequal load distribution. The paths, that receive excess traffic, will experience packet loss earlier than more evenly distributed paths, if we linearly increase the bandwidth utilization of the paths.

In the first stream group tests of Table 26, similar results were observed as earlier. Load balancing works well with different streams that have different bit rate. In the last test, a single flow burst of 9Mb/s average load was possible to transfer while several constant bit rate streams with a total load of 35Mb/s were transmitted. Additionally, in Table 28, a single burst of 3000 packets and 100Mb/s average load (frame IDT=13280ns, burts IDT=1098477205ns, 75301.2 packets/s) was transmitted to full ECMP group of eight load balancing links. The number of flows was increased until all the packets were successfully transmitted. 10 flows provides good enough load balancing to exceed momentarily the maximum bandwidth of 80Mb/s. The possible reason for this is the router's buffering in addition to load balancing. Sudden changes of traffic can be smoothed with small additional buffering.

7.2 Fast protection results and analysis

The physical layer used in test was a traditional Ethernet 10/100BASE-TX. The result was calculated the same way as the other fast protection tests explained in section 6.6. The average recovery time of a single ECMP link using 10/100BASE-TX was about 90ms. Other fast protection results are shown in Table 29. There is not much difference between the static routes using BFD and IGP using BFD. Only the software part, that is related to failure detection time, is relevant, when a failed link is directly connected to router's interface and ECMP paths are already

Table 28: Single Burst Packet Loss Test with Different Number of Flows and Eight ECMP Links

number of flows	packets received	% of received packets
1 flow	541 packets	$18,0\%$
2 flows	1168 packets	$38,9\%$
3 flows	1914 packets	$63,8\%$
4 flows	2812 packets	$93,7\%$
5 flows	2409 packets	$80,3\%$
6 flows	2274 packets	$75,8\%$
7 flows	2601 packets	$86,7\%$
8 flows	2906 packets	$96,9\%$
9 flows	2990 packets	$99,7\%$
10 flows	3000 packets	$100,0\%$

installed on the forwarding layer. In other situations, the software is as fast as in the single path case, as mentioned in the ECMP architecture section.

The IGP results without using BFD are in the last two lines of the table. Recovery time of a half second is possible with the default values except the hello timers that were at their minimum values. With a little tweaking of other parameters, such as enabling immediate hellos in OSPF, it was possible to remove a few hundred milliseconds. By setting timer values too aggressively, links would very likely end up oscillating in a slightly larger network than this.

Table 29: Recovery Times with different BFD values in Static, IS-IS and OSPF Case

BFD interval	Static		IS-IS		OSPF
10ms, mul 3	31ms		36ms		36ms
15ms, mul 3	40ms		42ms		40ms
30ms, mul 3	66ms		69ms		68ms
50ms, mul 3	136ms		104ms		133ms
100ms, mul 3	261ms		219ms		248ms
300ms, mul 3	638ms		-		-
1000ms,mul 3	2623ms		-		-
no BFD, def,	-		503ms		507ms
no BFD, tweak,	-		384ms		274ms

8 Conclusion

In this thesis different IP and MPLS traffic engineering methods were presented. ECMP is one of the most general solution for IP traffic engineering, providing solid load balancing and fast protection performance especially when it is used in combination with IP fast reroute. Because fast protection is handled with IP-FRR, link weights can be optimized from load balancing point of view. BFD provides a fast failure detection part for fast protection. With BFD, ECMP provides easily on the order of tens of milliseconds failure recovery.

As the results and the literature prove, ECMP outperforms single path solutions and it is competitive with even more complex MPLS solutions. ECMP does not need any additional configuration and the only thing to take care of is the adjustment of the link metrics in order to create a sufficient number of load balancing paths.

Changes to adopt ECMP are relatively small to existing single path control plane software. Changes to RIB and FIB table manipulation and SPF computation are the most relevant parts from a software point of view.

The quality of the load balancing algorithm of the forwarding plane has the most significant effect on load balancing performance. Traffic distribution is not always even. The rule of power of two paths seems to work, when the goal is even distribution. Additionally, the number of flows should be large enough in order to achieve efficient load balancing with flow-based algorithms. When hash-based algorithms are used, it is important to take care of the Traffic Polarization Effect and mitigate its influence on load balancing performance.

A simple, flow-based hash-threshold provides fairly good performance cost-efficiently. More sophisticated algorithms optimize jointly load balancing and packet reordering. One of the best algorithms is FLARE that achieves almost zero packets out-of-order and it still provides top load balancing performance and implementable solution and utilizes any given splitting vector effectively. TCP is evolving towards completely immune packet reordering. If this is achieved, less complex packet-based load balancing algorithms would outperform existing solutions.

References

[1] Rekhter, Y., Li, T., Hares, S., *A Border Gateway Protocol 4 (BGP-4)* IETF Request for Comments: 4271, 2006.

[2] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J. *Address Allocation for Private Internets* IETF Request for Comments: 1918, 1996.

[3] Fuller, V., *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan* IETF Request for Comments: 4632, 2006.

[4] Baker, F., *Requirements for IP Version 4 Routers* IETF Request for Comments: 1812, 1995.

[5] Postel, J., *Internet Control Message Protocol* IETF Request for Comments: 792, 1981.

[6] Information Sciences Institute University of Southern California 4676 Admiralty Way *Internet Protocol* IETF Request for Comments: 791 , 1981.

[7] Plummer, David C., *An Ethernet Address Resolution Protocol* IETF Request for Comments: 826, 1982.

[8] Bush, R., Meyer, D., *Some Internet Architectural Guidelines and Philosophy* IETF Request for Comments: 3439, 2002.

[9] Braden, R., *Requirements for Internet Hosts – Communication Layers* IETF Request for Comments: 1122, 1989.

[10] Moy, J., *OSPF Version 2* IETF Request for Comments: 2328, 1998.

[11] Oran, D., *OSI IS-IS Intra-domain Routing Protocol* IETF Request for Comments: 1142, 1990.

[12] Callon, R., *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments* IETF Request for Comments: 1195, 1990.

[13] Katz, D., Saluja, R., *Three-Way Handshake for IS-IS Point-to-Point Adjacencies* IETF Request for Comments: 5303, 2008.

[14] Balay, R., Katz, D., Parker, J., *IS-IS Mesh Groups* IETF Request for Comments: 2973, 2000.

[15] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., Xiao, X., *Overview and Principles of Internet Traffic Engineering* IETF Request for Comments: 3272, 2002.

[16] Nadas, S., *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6* IETF Request for Comments: 5798, 2010.

[17] Bryant, S., Pate, P., *Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture* IETF Request for Comments: 3985, 2005.

[18] Jacobson, V., Braden, R., Borman, D., *TCP Extensions for High Performance* IETF Request for Comments: 1323, 1992.

[19] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., McManus, J. *Requirements for Traffic Engineering Over MPLS* IETF Request for Comments: 2702, 1999.

[20] Deering, S., Hinden, R., *Internet Protocol, Version 6 (IPv6) Specification* IETF Request for Comments: 2460, 1998.

[21] Rajahalme, J., Conta, A., Carpenter, B., Deering, S., *IPv6 Flow Label Specification* IETF Request for Comments: 3697, 2004.

[22] Thaler, D., Hopps, C., *Multipath Issues in Unicast and Multicast Next-Hop Selection* IETF Request for Comments: 2991, 2000.

[23] Hopps, C., *Analysis of an Equal-Cost Multi-Path Algorithm* IETF Request for Comments: 2992, 2000.

[24] Atlas, A., Zinin, A., *Basic Specification for IP Fast Reroute: Loop-Free Alternates* IETF Request for Comments: 5286, 2008.

[25] Andersson, L., Minei, I., Thomas, B., *LDP Specification* IETF Request for Comments: 5036, 2007.

[26] Katz, D., Kompella, K., Yeung, D., *Traffic Engineering (TE) Extensions to OSPF Version 2* IETF Request for Comments: 3630, 2003.

[27] Smit, H., *Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE)* IETF Request for Comments: 3784, 2004.

[28] Mannie, E., Papadimitriou, D., *Recovery (Protection and Restoration) Terminology for Generalized Multi-Protocol Label Switching (GMPLS)* IETF Request for Comments: 4427, 2006.

[29] Lang, J.P., Rekhter, Y., Papadimitriou, D., *RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery* IETF Request for Comments: 4872, 2007.

[30] L. Berger, I. Bryskin, D. Papadimitriou, A. Farrel *GMPLS Segment Recovery* IETF Request for Comments: 4873, 2007.

[31] Lang, J., Rajagopalan, B., Papadimitriou, D., *Generalized Multi-Protocol Label Switching (GMPLS) Recovery Functional Specification* IETF Request for Comments: 4426, 2006.

[32] Deering, S., *ICMP Router Discovery Messages* Request for Comments: 1256, 1991

[33] Pan, P., Swallow, G., Atlas, A., *Fast Reroute Extensions to RSVP-TE for LSP Tunnels* IETF Request for Comments: 4090, 2005.

[34] Kompella, K., Rekhter, Y., Berger, L., *Link Bundling in MPLS Traffic Engineering (TE)* IETF Request for Comments: 4201, 2005.

[35] Vasseur, JP., Leroux, JL., Yasukawa, S., Previdi, S., Psenak, P., Mabbey, P., *Routing Extensions for Discovery of Multiprotocol (MPLS) Label Switch Router (LSR) Traffic Engineering (TE) Mesh Membership* IETF Request for Comments: 4972, 2007

[36] Katz, D., Ward, D., *Bidirectional Forwarding Detection (BFD)* IETF Request for Comments: 5880, 2010.

[37] Malkin, G., *RIP Version 2* IETF Request for Comments: 2453, 1998.

[38] Vasseur, JP., Ayyangar, A., Zhang, R., *A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)* IETF Request for Comments: 5152, 2008.

[39] Andersson, L., Asati, R., *Multiprotocol Label Switching (MPLS) Label Stack Entry: "EXP" Field Renamed to "Traffic Class" Field.* IETF Request for Comments: 5462, 2009.

[40] Bhandarkar, S., Reddy, A., Allman, M., Blanton, E., *Improving the Robustness of TCP to Non-Congestion Events* IETF Request for Comments: 4653, 2006.

[41] Allman, M., Paxson, V., Blanton, E., *TCP Congestion Control* IETF Request for Comments: 5681, 2009.

[42] Awduche, D., Berger, L., Li, T., Srinivasan, V., Swallow, G., *RSVP-TE: Extensions to RSVP for LSP Tunnels.* IETF Request for Comments: 3209, 2001.

[43] Kou, Z., Feng, L., *Update to OSPF Hello procedure, draft-kou-ospf-immediately-replying-hello-01.txt* IETF Internet Draft, 2006.

[44] Ogier, R., *OSPF Database Exchange Summary List Optimization* IETF Request for Comments: 5243, 2008.

[45] Walton, D., Retana, A., Chen, E., Scudder, J., *Advertisement of Multiple Paths in BGP, draft-ietf-idr-add-paths-04.txt* IETF Internet Draft, 2010.

[46] Zuo, Y., Pitts, J. *Impact of Link Weight Ranges on OSPE Weight Solutions* IEEE CHINACOM, 2007.

[47] Bhatia, M., Jakma, P., *Advertising Equal Cost Multipath routes in BGP, draft-bhatia-ecmp-routes-in-bgp-02.txt* IETF Internet Draft, 2006.

[48] Stein, Y., Insler, R., *PW Bonding, draft-stein-pwe3-pwbonding-01.txt* IETF Internet Draft, 2008.

[49] Bryant, S., Filsfils, C., Drafz, U., Kompella, V., Regan, J., Amante, S. *Flow Aware Transport of Pseudowires over an MPLS PSN, draft-ietf-pwe3-fat-pw-05* IETF Internet Draft, 2010.

[50] Villamizar, C. *OSPF Optimized Multipath (OSPF-OMP), draft-ietf-ospf-omp-02* IETF Internet Draft, 1999.

[51] Rajagopal, M., Rodriguez, E., Weber, R. *Fibre Channel Over TCP/IP (FCIP)* IETF Request for Comments: 3821, 2004.

[52] Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M., Zeidner, E., *Internet Small Computer Systems Interface (iSCSI)* IETF Request for Comments: 3720, 2004.

[53] Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., Pillay-Esnault, P. *Multi-Topology (MT) Routing in OSPF* IETF Request for Comments: 4915, 2007.

[54] Stewart, R., *Stream Control Transmission Protocol* Request for Comments: 4960, 2007

[55] Przygienda, T., Sagl, Z., Shen, N., Sheth, N., Pillay-Esnault, P. *M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)* IETF Request for Comments: 5120, 2008.

[56] Fransson, P. Carr-Motyckova, L. *Loop-Free Link-State Routing* Computer Communications and Networks ICCCN, pp. 905–911., 2007

[57] Francois, P., Bonaventure, O. *Avoiding Transient Loops During the Convergence of Link-State Routing Protocols* IEEE/ACM Transactions On Networking, VOL. 15, NO. 6., 2007

[58] Wang, Y. Wang, Z. Zhang, L. *Internet Traffic Engineering without Full Mesh Overlaying* IEEE INFOCOM vol 1. pp. 565–571, 2001.

[59] Lin, W., Liu, B., Tang, Y., *Traffic Distribution over Equal-Cost-Multi-Paths using LRU-based Caching with Counting Scheme* IEEE AINA, 2006.

[60] Handley, M., Raiciu, C., Barre S., Iyengar, J., *Architectural Guidelines for Multipath TCP Development, draft-ietf-mptcp-architecture-03,* IETF Internet-Draft, 2010.

[61] Handley, M., Raiciu, C., Ford, A. *TCP Extensions for Multipath Operation with Multiple Addresses draft-ietf-mptcp-multiaddressed-02,* IETF Internet-Draft, 2010.

[62] Atlas, A., *U-turn Alternates for IP/LDP Fast-Reroute, draft-atlas-ip-local-protect-uturn-03* Internet-Draft, 2006.

[63] Bryant, S., Shand, M. *IP Fast Reroute using tunnels, draft-bryant-ipfrr-tunnels-03*IETF Internet-Draft, 2007

[64] Villamizar, C., *Use of Multipath with MPLS-TP and MPLS draft-villamizar-mpls-tp-multipath-00* IETF Internet-Draft, 2010

[65] Kompella, K., *Multi-path Label Switched Paths Signaled Using RSVP-TE draft-kompella-mpls-rsvp-ecmp-00.txt* IETF Internet-Draft, 2010

[66] Handley, M., Raiciu, C., Wischik, D. *Coupled Multipath-Aware Congestion Control draft-ietf-mptcp-congestion-00,* IETF Internet-Draft, 2010.

[67] Dreibholz, T., Becke, M., Iyengar, J., Natarajan, P., Tuexen, M., *Load Sharing for the Stream Control Transmission Protocol (SCTP) draft-tuexen-tsvwg-sctp-multipath-01.txt* IETF Internet-Draft, 2010

[68] Youjun, B., hong, G., Hongchao, H., Binqiang, W., *A Traffic Splitting Algorithm Based on Dual Hash Table for Multi-path Internet routing* IEEE MVHI, 2010.

[69] Martin, R., Menth, M., Hemmkeppler, M., *Accuracy and Dynamics of Hash-Based Load Balancing Algorithms for Multipath Internet Routing.* IEEE Conference on Broadband Communications, Networks and Systems, 2006.

[70] Zinin, A., *Cisco IP Routing, Packet Forwarding and Intradomain Routing Protocals,* vol. Section 5.5.1. Addison Wesley, 2002.

[71] Kandula, S., Katabi, D., Sinha, S., Berger, A., *Dynamic Load Balancing Without Packet Reordering* ACM SIGCOMM Computer Communication Review 54 Volume 37, Number 2, 2007.

[72] Aly, S. A., Ansari, N. Walid, A. I., *Secure Coding-based Multipath Adaptive Traffic Engineering* IEEE, 2010.

[73] Fujinoki, H., Ansari, N. Walid, A. I., *Multi-Path BGP (MBGP): A Solution for Improving Network Bandwidth Utilization and Defense against Link Failures in Inter-Domain Routing* IEEE ICON, 2008.

[74] Balon, S., Skivee, F., Leduc, G., *How Well do Traffic Engineering Objective Functions Meet TE Requirements?* IFIP Networking, LNCS 3976, pp. 75–86, 2006.

[75] Fortz, B., Thorup, M., Leduc, G., *Robust optimization of OSPF/IS-IS weights* International Network Optimization Conference, 2003.

[76] Rusu, O., Vraciu, V., *IS-IS metric optimization* IEEE International Conference, 2010.

[77] Leung, K., Li, V., Yang, D., *An Overview of Packet Reordering in Transmission Control Protocol (TCP): Problems, Solutions, and Challenges* IEEE Transections on Parallel and Distributed Systems, Vol. 18, No. 4, 2007.

[78] Cao, Z., Wang, Z., Zagura, E., *Performance of Hashing Based Schemes for Internet Load Balancing* IEEE INFOCOM, 2000.

[79] Leduc, G., Abrahamsson, H. Balon, S. et al., *An open source traffic engineering toolbox Computer Communications* 29 pp. 593–610., 2006.

[80] , Kvalbein, A., Lysne, O., *How can Multi-Topology Routing be used for Intradomain Traffic Engineering?* SIGCOMM workshop on Internet network management, 2007.

[81] Labovitz, C., Malan, G. R., Jahanian, F. *Internet Routing Instability* IEEE/ACM Transactions on Networking, Vol. 6. No. 5., 1998.

[82] Abrahamsson, H. Bjorkman, M., *Robust traffic engineering using l-balanced weight-settings in OSPF/IS-IS* Broadband Communications, Networks, and Systems, 2009.

[83] Lee, Y., Park, I, Choi, Y., *Improving TCP Performance in Multipath Packet Forwarding Networks* Journal of Communications and Networks, VOL.4, NO.2, 2002.

[84] *IGP-WO*, Web site, http://www.poms.ucl.ac.be/totem/, 2 March 2011

[85] Thorup, M., *Increasing Internet Capacity Using Local Search*, Computational Optimization and Applications, 29 pp. 13–48, Fortz, B., 2004.

[86] Fortz, B., Thorup, M., *nternet traffic engineering by optimizing OSPF weights*, IEEE INFOCOM, pp. 519–528, 2000.

[87] Pióro, M., Szentesi, À., Harmatos, J. et al., *On open shortest path first related network optimisation problems.* Performance Evaluation 48 pp. 201–223, 2002.

[88] Ericsson, M., Resende, M. G. C., Pardalos, P. M., *A genetic algorithm for the weight setting problem in OSPF routing*, Journal of Combinatorial Optimization Volume 6, Number 3, pp. 299–333, 2002.

[89] Buriol, L.S., Resende, M.G.C., Ribeiro, C. C., Thorup, M., *A Hybrid Genetic Algorithm for the Weight Setting Problem in OSPF/IS-IS Routing.* Wiley Periodicals, Inc. Networks, Vol. 46(1), pp. 36–56, 2005.

[90] Michael, G., Schneider, G., Nemeth, T., *A simulation study of the OSPF-OMP routing algorithm.*, Computer Networks 39 pp. 457–468., 2002.

[91] Chim, T., Yeung, L., Lui, K. *Traffic distribution over equal-cost-multi-paths,* Computer Networks 49, pp. 465–475, 2005.

[92] Tian, M., Lan, J., Zhu, X., Huang, J., *A Routing Optimization Algorithm of Equal-Cost-Multi-Paths Based on Link Criticality* IEEE, 2010.

[93] Xu, D., Chiang, M., Rexford, J. *Link-State Routing with Hop-by-Hop Forwarding Can Achieve Optimal Traffic Engineering* IEEE INFOCOM, pp. 466–474, 2008.

[94] Laor, M., Gendel, L., *The effect of packet reordering in a backbone link on application throughput.* IEEE Network, 2002.

[95] Guerin, R., *QoS Routing Mechanisms and OSPF Extensions,* IEEE GLOBECOM pp. 1903–1908., 1997.

[96] Wang, Z., *Quality of Service Routing for Supporting Multimedia Applications,* IEEE JSAC, Vol. 14, No. 7, pp. 1228–1234., 1996

[97] Riedl, A., Schupke, D., *Routing Optimization in IP Networks Utilizing Additive and Concave Link Metrics.* IEEE/ACM Transactions on Networking, Vol. 15, No. 5, 2007.

[98] Prabhavat, S., Nishiyama, H., Ansari, N., Kato, N., *On the Performance Analysis of Traffic Splitting on Load Imbalancing and packet Reordering of Bursty Traffic.* IEEE Proceedings of IC-NIDC, 2009.

[99] Xi, K., Chao, H., *ESCAP: Efficient SCan for Alternate Paths to Achieve IP Fast Rerouting* IEEE GLOBECOM, 2007.

[100] Kandula, S., Katabi, D., Davie, B., Charny, A., *Walking the Tightrope: Responsive Yet Stable Traffic Engineering* ACM SIGCOMM, 2005.

[101] Wang H., Xie, H., Qiut, L., *COPE: Traffic Engineering in Dynamic Networks* ACM SIGCOMM, 2006.

[102] Kar, K., Kodialam, M., Lakshman, T., *Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications* IEEE Journal on selected areas in communications, Vol. 18, No. 12., 2000.

[103] Zhu, M, Ye ,B., Feng S., *A new dynamic routing algorithm based on minimum interference in MPLS Networks* IEEE WICOM, 2008.

[104] Li, Y., Harms, J., Holte, R., *A Simple Method for Balancing Network Utilization and Quality of Routing* IEEE ICCCN, 2005.

[105] Elwalid, A., Jin, C., Low, S., Widjaja, I., *MATE: MPLS Adaptive Traffic Engineering* IEEE INFOCOM, 2001.

[106] Boutaba, R., Szeto, W., Iraqi, Y., *DORA: Efficient Routing for MPLS Traffic Engineering,* Journal of Network and Systems Management Volume 10, Number 3, pp. 309–325, 2002.

[107] Gjoka, M., Ram, V., Yang, X., *Evaluation of IP Fast Reroute Proposals* IEEE, 2007.

[108] Szilàgyi, P., Tòth, Z., *Design, Implementation and Evaluation of an IP Fast ReRoute Prototype.* Technical report, BME, 2008.

[109] Freeman, Garey, M., Johnson, D., *Computers and Intractability: A Guide to the Theory of NPCompleteness,* 1979. ISBN: 0-7167-1045-5

[110] Čičič, T., Hansen, A., Kvalbein, A. et al., *Relaxed Multiple Routing Configurations: IP Fast Reroute for Single and Correlated Failures* IEEE Transacctions on Network and service management, Vol. 6, No. 1, 2009.

[111] Enyedi, G., Szilàgyi, P., Rètvàri, G. et al., *IP Fast ReRoute: Lightweight Not-Via without Additional Addresses* IEEE INFOCOM, 2009.

[112] S., Lee, S., Yu, Y., Zhang, Z.-L., and Chuah, C-N., *Fast Local Rerouting for Handling Transient Link Failures.* IEEE/ACM Transactions on Networking, Nelakuditi, 2006.

[113] Paxson, V., *End-to-End Internet Packet Dynamics,* ACM SIGCOMM, 1997.

[114] Gharai, L., Perkins, C., Lehman, T., *Packet Reordering, High Speed Networks and Transport Protocol Performance*, IEEE ICCCN, 2004.

[115] Shand, M., Bryant, S., Previdi, S., *IP Fast Reroute Using Not-via Addresses, draft-ietf-rtgwg-ipfrr-notvia-addresses-05* IETF Internet-Draft, 2010.

[116] Kompella, K., Amante, S., *The Use of Entropy Labels in MPLS Forwarding draft-kompella-mpls-entropy-label-01* IETF Internet-Draft, 2010.

[117] Kamamura, S., Miyamura, T., Pelsser, C. et al., *Scalable Backup Configurations Creation for IP Fast Reroute* IEEE, 2009.

[118] Čičič, T., Hansen, A., Kvalbein, A. et al., *Fast IP Network Recovery using Multiple Routing Configurations* IEEE INFOCOM, 2006.

[119] Martin, R., Menth, M., Hemmkeppler, M. *Accuracy and Dynamics of Multi-Stage Load Balancing for Multipath Internet Routing* IEEE ICC, 2007.

[120] Gojmerac, I. Ziegler, T. Ricciato, F. Reichl, P., *Adaptive multipath routing for dynamic traffic engineering*, IEEE GLOBECOM Volume: 6, pp. 3058–3062, 2003.

[121] Sundaresan, S., Lumezanu, C., Feamster, N., *Autonomous traffic engineering with self-configuring topologies*, ACM SIGCOMM, 2010.

[122] Farkas, K., *Ip traffic engineering using omp technique*, IASTED. PDCS, 2000.

[123] Rost, S., Balakrishnan, H., *Rate-aware splitting of aggregate traffic.*, Technical report. MIT, 2003.

[124] *Cariden's software*, Web site, http://www.cariden.com/products/functional_diagram/ , 2 March 2011

[125] *Wandl IP/MPLSView*, Web site, http://www.wandl.com , 2 March 2011

[126] *Opnet SP Guru*, Web site, http://www.opnet.com , 2 March 2011

[127] Srivastava, S., Agrawal, G., Pï¿½oro, M., Medhi, D., *Determining Link Weight System under Various Objectives for OSPF Networks using a Lagrangian Relaxation-Based Approach*IEEE Transactions on Network and Service Management, Vol. 2, No. 1, 2005

[128] Moy, J. T., *OSPF: Anatomy of an Internet Routing Protocol* , Addison Wesley., 1998. ISBN: 0-201-63472-4.

[129] Alwayn, V., *Advanced MPLS Design and Implementation* Second Edition. Cisco Press., 2003. ISBN: 1-58705-020-X.

[130] Minei, I., Lucek, J., *MPLS-Enabled Applications: Emerging Developments and New Technologies* 2nd Edition. John Wiley & Sons, 2008. ISBN: 978-0-470-98644-8.

[131] Viswanathan, A., Rosen, E., Callon, R., *Multiprotocol Label Switching Architecture* IETF Request for Comments: 3031, 2001.

[132] Clark, M. P., *Data Networks, IP and the Internet: Protocols, Design and Operation* John Wiley & Sons, 2003. ISBN: 0-470-84856-1.

[133] Doyle, J. *OSPF and IS-IS: Choosing an IGP for Large-Scale Networks* Addison Wesley., 2005. ISBN: 0-321-16879-8.

[134] Gredler, H., Goralski, W., *The Complete IS-IS Routing Protocol* Springer. 2005. ISBN: 1-85233-822-9.

[135] Skiena, S., *Shortest Paths. 6.1 in Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica.* Addison-Wesley, pp. 225–253, 1990.

[136] Bellman, R., *On a Routing Problem* Quarterly of Applied Mathematics, 16(1), pp. 87–90, 1958.

[137] Wang, N., Ho, K., Pavlou, G., Howarth, M, *An Overview of Routing Optimization for Internet Traffic Engineering* IEEE Communications vol 10, No. 1., 2008.

[138] Fredman, M.L. Willard, D.E., *Trans-dichotomous algorithms for minimum spanning trees and shortest paths.* J. Comp. Syst. Sc. 48, pp. 533–551, 1994.

[139] Dijkstra, E.W., *A note on two problems in connection with graphs.* Numerische Mathematik, Volume 1, pp. 269–271, 1959.

[140] Fredman M. L., Tarjan R. E., *Fibonacci heaps and their uses in improved network optimization algorithms.* Journal of the ACM 34(3), 596–615., 1987.

[141] Leon-Garcia, A., Widjaja, I., *Communication Networks, Fundamental Concepts and Key Architectures.* Second Edition. The McGraw Companies, 2004. ISBN: 0-07-246352-X

[142] Huitema, C., *Routing in the Internet,* 2nd Edition. Prentice Hall, Upper Saddle River, 2000. ISBN: 0-13-022647-5.

[143] International Standard ISO/IEC 7498–1, *Information Technology - Open Systems Interconnection - Basic Reference Model: The Basic Model* Second Edition. , 1996.

[144] International Standard ISO/IEC 10589:2002(E), (ISO 8473) *Information Technology - Telecommunications and information exchange between systems - Intermediate System to Intermediate System intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service* Second Edition. , 2002.

[145] Comer, D. E., *Internetworking with TCP/IP, Volume I: Principles, Protocols, and Architecture,* Fifth Edition. Prentice Hall International, 2006. ISBN: 0-13-187671-6.

[146] Cormen, T. H., Leiserson, C. E., Rivest, R. L., *Introduction to Algorithms.* The MIT Press, Cambridge, Massachusetts, 2000. ISBN: 0-262-53091-0.

[147] Tanenbaum, A. S., *Computer Networks,* Third Edition. Prentice Hall International, Upper Saddle River, 1996. ISBN: 0-13-394248-1.

A Appendix

Table A1: Command Description

bold	Bold text indicates words, which are literally written as shown. italics Italic text indicates argument for which a value must be supplied.
|	A vertical line indicates a choice of values. The symbol is not part of the command, i.e. not typed when typing the CLI command.
$[x \mid y]$	Square brackets indicate optional value(s). These symbols are not part of the command, i.e. not typed when typing the CLI command.
$\{x \mid y\}$	Braces indicate mandatory value(s). These symbols are not part of the command, i.e. not typed when typing the CLI command.

Table A2: Convention Description

global	Global routing table.
recursive	Permit recursive next hop.
recursive-mpls	Use MPLS LSP to reach the specified next hop.
A.B.C.D	Specifies the IP destination prefix. Range: Any legal IPv4 address.
M	Specifies the IP destination prefix mask length as bits. Range: 0 .. 32
dest-vrf	Specifies the destination VRF. Range: Any existing VRF name.
distance	Specifies the distance value for the route. Range: 1 .. 255 Default Value: 1
gateway-ip	Specifies the IP gateway address (peer interface address). Range: Any legal IPv4 address.
interface	Interface into which packets are routed. Range: Any existing interface supporting IP routing