NTNU
Innovation and Creativity

Measurement Analysis and Improvement of rerouting in UNINETT

Håkon Skaland

Master of Science in Communication Technology
Submission date: June 2007
Supervisor: Bjarne Emil Helvik, ITEM
Co-supervisor: Trond Skjesol, UNINETT

Norwegian University of Science and Technology
Department of Telematics

Problem Description

UNINETT has been measuring packet loss in connection with link and router failures. Packet loss is experienced both during change in the preferred routing path and rearranging back to the normal state. Packet loss during handling of link and router failures implies downtime even though the network is physically connected. These downtimes are a significant contributor to the service unavailability and reduction of the overall Quality of Service (QoS). It is of interest to analyse the measurements to see how it is possible to use this improved view of the network to increase the QoS.

The requirements for network availability is getting higher, and to follow this trend it is necessary to reduce the rerouting times and periods with packet loss. The rerouting time is the downtime from the start of the incident causing a change in the routing path, to the time when the network has converged and is stable. In UNINETT, which is a network with fault handling by path restoration and a Link State routing protocol (IS-IS), this downtime has three contributors:

1. Time to detect failure.

2. Delay before failure is considered permanent.

3. Convergence time.

Micro loops in the network can give longer convergence time and cause packet loss. This happens both when a failure causes a change in the current best path, and when a repair changes the routing path back to the normal state. Micro loops are relatively easy to eliminate in a network with full mesh connectivity, but UNINETT's topology has a ring structure.

The objective of this assignment is, through analysis of the mentioned measurements, to get an improved understanding of how the various contributors affect the end-to-end downtime and unavailability. On this basis suggest a scheme for improvement of the fault handling.

The motivation for UNINETT is to make the network reroute faster in the case of a failure. A faster rerouting can reduce the packet loss, which improves the QoS. The goal for this report is to suggest remedial actions to improve the rerouting times in UNINETT.

The following tasks are foreseen:

- Study the fault handling in networks with path restoration techniques. Here under the study includes important phenomena like micro loops which may impact the overall QoS.
- Determine the effect of the downtime contributors on the network's end-to-end characteristics.
- Examine how the parameters in IS-IS affect the overall routing scheme.
- Propose alternative parameterizations or a procedure towards finding such parameterizations that improves the restoration performance in UNINETT.
- Study the topology and the cost metrics in UNINETT. Propose changes to improve the rerouting times.
- Investigate the possibility to implement a technique in UNINETT for earlier detection of failures.
- To prove the results, the solutions should be prepared for testing. The test can either be a simulation, a test bed, or "in real" in the UNINETT's production network. If there is willingness and free resources within UNINETT, and there is enough time within the deadline of the project, the test will be accomplished.

The current rerouting time in UNINETT is about 7 seconds. The expected deliverable of this work is a report documenting the solutions to improve the rerouting times in UNINETT. A decrease in the rerouting times will possibly reduce the packet loss, that in turn will increase the QoS.

Assignment given: 17Th of January 2007.
Supervisor: Bjarne Emil Helvik, ITEM.

Measurement Analysis and Improvement of rerouting in UNINETT

Håkon Skaland
Department of Telematics
Norwegian University of Science and Technology

June 14, 2007

Preface

This master thesis is my final paper in the 2-year MSc study at the Department of Telematics[1] at the Norwegian University of Science and Technology[2] in Trondheim. I have a BSc degree in Electrical Engineering from Sør-Trøndelag University College[3], which made me qualified for the 2-year MSc study. The thesis gives a credit load equivalent to 30 ECTS[4] Credits, and the project duration was 20 weeks. The work with the project has been performed at UNINETT's office in Abels gt. 5 in Trondheim in the spring of 2007. The picture on the front page is taken at the UNINETT office and shows the test lab of the project. My co-supervisor, Trond Skjesol at UNINETT, proposed the outline of the assignment. With the help from my supervisor, Bjarne Emil Helvik at ITEM, the final problem description was defined 1 month after the start of the project.

I would like to thank Bjarne Emil Helvik and Trond Skjesol for their help and support through the project. I would also like to thank the rest of the staff at UNINETT for their technical support, and for taking good care of me and treating me like an colleague. There were two other students from ITEM who carried out their master thesis for UNINETT simultaneously with me; Vetle Toreid and Helene Abrahamsen. I would like to thank both of them for the technical cooperation and discussions we have had during the work with our projects. A thanks goes also to my good friends, Bjørn Ove Kvello and Børre Johansen, who read the thesis and gave me feedback before the deadline of the project. Last, but not least, I would like to thank my girlfriend Dagmar for helping me finish my studies with her love and support.

Trondheim, June 14Th 2007

Håkon Skaland

[1] ITEM
[2] NTNU
[3] HiST
[4] European Credit Transfer System

Abstract

This thesis focuses on the analysis of the rerouting times in UNINETT, the Norwegian research network. Rerouting happens in case of a topology change of the network, and the routers have to calculate new paths to all destinations. The downtimes due to rerouting is a major contributor to the overall service unavailability. Because of this it is of interest to study the different components of these downtimes, and propose changes to speed up the rerouting process. The main goal for the thesis is to improve the service availability in UNINETT. In UNINETT there are measurements of periods of packet loss available. These measurements, as well as the statistics from the network nodes, are analysed and the results are presented in this thesis.

UNINETT is a network with IS-IS as the routing protocol, and fault handling by path restoration. Fault handling by path restoration means there is no spare capacity to switch to in case of a link or node failure. All routers in the network have to be updated about the topology change, and find new paths around the failure. The delay to update the network nodes to a common stable view, is called the convergence time. During this period it is observed packet loss, and it is of interest to make this period as short as possible. The reason for the packet loss during the convergence time is the inconsistency in the router's routing and forwarding tables. The construction of transient loops between nodes can happen in this phase. This will impose extra load to the network, delay, and in worst case loss of packets. These loops are called micro loops and increase the downtime during the convergence period.

The parametrization in IS-IS is studied, and changes to the parameter values are proposed. Too much tuning of the parameters may introduce instability in the network, which increase the load to the nodes and links. This can lead to even longer convergence time, and periods with packet loss. The recommended values are tested in a small test lab replicating parts of the topology of Northern Norway in UNINETT. The results from the test are compared with a case study of a failure on the Trondheim-Tromsø link in UNINETT. The observations from the case study show a typical delay of up to 10 s for the convergence time. The results from the test lab show that it is possible to achieve sub-second convergence time, without any compromise on the stability of the network. Due to the small scale of the test lab, the traffic intensity was too low to observe any overload to the nodes or links. This may be a problem in a full scale network like UNINETT, and further testing is recommended before the proposed changes to

the parameters are implemented in any production network.

The fault handling in UNINETT is also studied, which includes the contribution from the different components to the convergence time. The observations and results from the mentioned case study and test lab are used in this study too. It is observed that the timer delay before the "Shortest Path Tree" computation is run in the routers, is the major contributor to the convergence time. This is improved by tuning the SPT timers, as observed in the results from the test lab. The failure detection and flooding components are also large contributors to the convergence time. The test lab shows that the failure detection is improved by tuning the hello parameters in IS-IS, but a less processor intensive method called BFD is recommended for further study. The parameters triggering the data-link layer timers may also be a possibility to speed up the failure detection, but this is not further investigated in the thesis. The flooding component is reduced by enabling the fast flooding command.

The phenomena of micro loops is studied, and the method called oFIB is recommended for implementation in UNINETT. Micro loops are a small contributor to the convergence time in UNINETT today, but the customers' requirements for service availability are increasing, and the necessity of a solution is in near future. In addition to the oFIB method, a fast repair technique like IPFRR may eliminate almost all downtimes during rerouting in UNINETT, but this is subjects for further studies.

General Terms:

Measurement, analysis, rerouting, testing, fault handling.

Keywords:

IS-IS parametrization, convergence time, UNINETT, micro loop.

Contents

List of Figures

List of Tables

Abbreviations

Ack — Acknowledge

ARP — Address Resolution Protocol

AS — Autonomous System

CLNP — ConnectionLess Network Protocol

CLNS — ConnectionLess Network Service

Conv — Convergence

CPU — Central Processing Unit

CSNP — Complete Sequence Number Packet

DHCP — Dynamic Host Communication Protocol

DIS — Designated Intermediate System

ES-IS — End System-to-Intermediate System

FIB — Forward Information Base

FRR — Fast ReRoute

ICMP — Internet Control Message Protocol

IETF — Internet Engineering Task Force

IGP — Interior Gateway Protocol

IIH — IS-IS Hello

IP — Internet Protocol

IS-IS — Intermediate System-to-Intermediate System

ISH — Intermediate System Hello

ISO — International Organization for Standardization

ISP — Internet Service Provider

ITU — International Telecommunications Union

LAN — Local Area Network

LS — Link State

LSDB — Link State DataBase

LSP — Link State Packet

MAC — Media Access Control

MPLS — MultiProtocol Label Switching

MSC — Message Sequence Chart

MTU — Maximum Transmission Unit

NBMA — NonBroadcast MultiAccess

NSAP — Network Service Access Point

oFIB — Ordered FIB

OSI — Open System Interconnection

P2P — Point-To-Point

PDU — Protocol Data Unit

PLSN — Path Locking with Safe-neighbors

PRC — Partial Route Calculation

PSNP — Partial Sequence Number Packet

Retransm — Retransmission

RFC — Request For Comments

RIB — Routing Information Base

SDH — Synchronous Digital Hierarchy

SLA — Service Level Agreement

SNP — Sequence Number Packet

SNPA — SubNetwork Point of Attachment

SPF — Shortest Path First

SPT — Shortest Path Tree

SRM — Send Routing Message

SSN — Send Sequence Number

Synch — Synchronization

SysID — System Identifier

TCP — Transmission Control Protocol

TLV — Type, Length, and Value

ToS — Type of Service

UDP — User Datagram Protocol

VoIP — Voice over IP

Chapter 1

Introduction

This master thesis discuss different methods to reduce the downtime during rerouting in UNINETT's production network. UNINETT is, among other things, an ISP (Internet Service Provider) for universities, colleges, and research institutions in Norway, owned by the Norwegian Ministry of Education and Research[1].

This chapter describes the motivation and background for the thesis, what is hoped to accomplish, and the limitation for the work. At the end of the chapter is a guide to the rest of the thesis.

1.1 Motivation

The motivation for the thesis is to decrease the periods of packet loss after a topology change in UNINETT. The packet loss in these situations is due to the routers' inconsistent view of the topology of the network, and is a major contributor to the reduction of the overall QoS. The reason for the inconsistency is the delay from the failure occurs, to all routers have updated their routing and forwarding tables.

Real time applications like VoIP (Voice over IP) and video streaming have evolved in popularity, and have high QoS requirements. This has made the subject of IP resilience a hot topic in network research groups[2][3]. Because of this the subject of the thesis is very important for ISPs. An implementation of the suggested solutions in their production network may decrease the packet loss, thus increasing the QoS. An improved QoS is an advantage for the ISP in a marked where the customers are constantly demanding better availability of the services.

1.2 Background

Packets are lost in the period after a topology change, when the routers do not share a common view of the network. The reasons are the continuing at-

tempts to use the failed component, as well as forwarding loops. These transient loops, known as micro loops, arise due to the mentioned inconsistency in the routers' view of the topology of the network. Micro loops may arise when the failure occurs, and when it is repaired. Even a change to a link metric may introduce micro loops and periods of packet loss, without losing the physical connection. This is a problem in both IP networks and MPLS networks[4], and has led to the development of a fast reroute mechanism for MPLS[5]. Work is in progress within the IETF to specify alternative mechanisms that may be deployed in MPLS networks and in IP networks[6][7]. One of these mechanisms is called *IP fast reroute (IPFRR)*[8]. In case of a failure, IPFRR uses precalculated backup next-hops that are loop free and safe to use until the topology converge. In this way the downtime is minimized when a failure occurs. These fast reroute mechanisms are of little use if the disruptive effect of micro loops is not minimized, which led to the development of mechanisms to control the re-convergence process[9][10].

UNINETT are performing measurements of packet loss in the connection with link and node failures. These measurements have shown periods of packet loss typically up to 10 s when the link or node fails, and less than 4 s when it is repaired. The periods of packet loss is experienced as downtimes for the destinations trying to pass traffic through the failed component.

As far as known to the author there have not been performed measurements in this large scale in other networks, at least not with publicly available results. Because of that is the thesis and its results interesting for both research network and commercial networks, to compare the results with the situation in their networks.

1.3 Purpose and Goals

As described in the problem description; the purpose of the thesis is to get an improved understanding of how the different contributors of the rerouting time affect the end-to-end downtime and unavailability. The three contributors described in the problem description are "converted" to six components in Chapter 3, to adapt to common use in literature about the subject.

The goal is to use this understanding to suggest remedial actions to improve the rerouting times in UNINETT. As mentioned is the current rerouting times in UNINETT up to 10 s. Researchers have demonstrated in a simulator the possibility to achieve sub-second rerouting in large IP networks [11]. This research is elaborated in the thesis to show that sub-second downtime during rerouting is achievable in UNINETT.

What is new with the thesis is the level of details in the recommended parametrization, and the results from the suggested actions in a test lab.

1.4 Limitations

There are many different techniques in fault handling, dependent on the network technology. The different technologies include protection and restoration mechanisms in both IP and MPLS networks. The thesis limits the study to IP networks with path restoration in case of a failure. This means there are no spare protection link to switch to if a link fails. The network has to adapt to the change in the topology, and find new paths around the failure. UNINETT is an IP network with fault handling by path restoration, and is used as the base for the thesis.

A second limitation is the study of parametrization of routers, and the tuning of those. The thesis includes parameters controlling the IS-IS routing protocol in general. Parameters in Cisco routers are specially studied, and given recommended values. The thesis describes very short the possibility to speed up the failure detection by tuning the data-link layer parameters, like SDH or Ethernet. No testing of those parameters has been performed.

1.5 Guide to the Thesis

The thesis is organized as follows:

- Chapter 2 gives the theoretical background about the IS-IS routing protocol. This information is needed for the parametrization discussed in Chapter 5.4, which is used in the test lab described in Chapter 5.5.
- Chapter 3 gives the theoretical background needed about fault handling. The theory includes phrases and expressions used in the case study and the test lab in Chapter 5.
- Chapter 4 discuss the problem with micro loops, and describes possible solutions. These solutions are the background for some of the conclusions in Chapter 6.
- Chapter 5 describes the measurements available in UNINETT, and the methodology to analyse the rerouting times. The theory in Chapter 2 and 3 is used when comparing the observations from a case study in UNINETT and a test lab. The results from this comparison is the main part of the conclusions in Chapter 6.
- Chapter 6 summarize the work and results from the thesis, and draw conclusions. The chapter ends with a section about suggested further work.

The reference list includes some IETF drafts. All these drafts are available for download as a ZIP-file from the master thesis database[12] at NTNU.

Chapter 2

IS-IS

This chapter describes the IS-IS routing protocol: A brief summary of the background and history, the hierarchical building blocks, and the exchange of messages between the nodes. IS-IS is chosen as the interior routing protocol in UNINETT, which makes the information in this chapter important for the thesis.

It is meant as a quick introduction to those not familiar with IS-IS, and to give the theoretical background for the work performed in the thesis. Especially Section 2.3 is important for the case study and the test in Chapter 5. The information in this chapter is based on [13], and the default values refers to Cisco routers[14] if not else stated.

2.1 Introduction to IS-IS

IS-IS is an *Interior Gateway Protocol (IGP)* based on the *Link-State (LS)* technology. IGPs are used within an *Autonomous System (AS)* to build and maintain a consistent view of the topology of the network.

To build the adjacencies and to maintain the view of the network, the routers send different *Protocol Data Units (PDUs)* to each other and calculate a *Shortest Path First (SPF)* tree of the topology. A PDU contains information about the sender and information about the network, known to the sender. A PDU is also called a packet.

2.1.1 Background and History

Intermediate System-to-Intermediate System (IS-IS) is developed by the *International Organization for Standardization (ISO)*. The first edition was published in 1990. It is based on the seven layer reference model known as the *Open System Interconnection (OSI)* model[15]. The OSI model provides the architectural framework for developing open standards for interconnectivity and interoperability between communication equipments from different vendors. The OSI model

is illustrated in Figure2.1 were some of the expressions used in the thesis are pointed out.

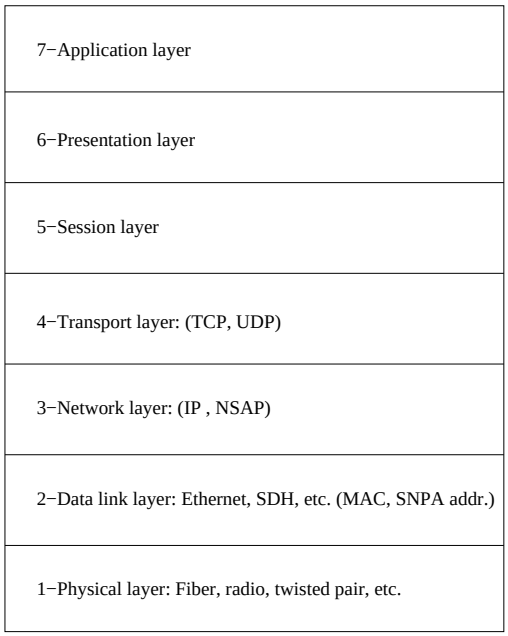

Figure 2.1: The 7 layer OSI model

Within this framework ISO specified two types of data communication services:

Connection Network Service (CONS) requires setup of the connection end-to-end prior of the data transmission.

Connectionless Network Service (CLNS) communicates hop-by-hop on the path to the destination.

IS-IS was designed to provide the necessary intelligence for automatic and dynamic routing of data packets in a CLNS environment. The IS-IS protocol is specified in *ISO 10589*[16]. The ISO standard was intended to be used in routers running *Connectionless Network Protocol (CLNP)*[17] in a CLNS environment.

ISO also specified the *End System-to-Intermediate System (ES-IS)*[18] routing exchange protocol for use in conjunction with CLNP. The ES-IS protocol was designed for host-to-router communication in a pure ISO environment. The intermediate system represents a router, and the end system represents the host. Due to the huge popularity and fast growing of IP networks, it was necessary to

adapt IS-IS to also work within the IP framework. IETF has specified a IS-IS version for the TCP/IP environment, commonly known as *Integrated IS-IS* or *Dual IS-IS*, in *RFC 1195*[19].

The ES-IS protocol in a CLNS environment has basically the same operation as a combined version of ICMP, ARP and DHCP within the IP framework. In a network that only supports IP, the ES-IS protocol is not needed for the processing and transfer of datagrams. Because the IS-IS protocol is dependent of some functions provided by the ES-IS protocol, the ES-IS protocol is still needed for background support.

2.2 Levels in IS-IS

The IS-IS protocol supports a two level hierarchy to split networks in different areas; Level 1 and Level 2. A network can consist of just Level 1, Level 2, or both levels. Level 1 routing involves the local area, and Level 2 routing connect the backbone together.

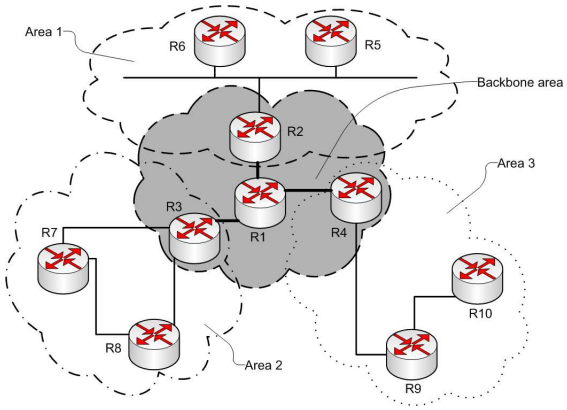

Figure 2.2: The hierarchically network levels and areas in IS-IS.

To establish an adjacency between neighbors, the routers must have configured the same level type or Level 1-2 on the link. Each router uses Dijkstra's algorithm to calculate a SPF tree with itself as root, and sends its view of the area to the neighbors. If an area consists of many routers, the SPF calculation and PDU exchange will be very processor intensive and bandwidth demanding. The advantage of splitting the network in different areas is to manage and scale routing in large networks. The areas limit the size of the topology as seen from each router, which limit the flooding and SPF calculation.

Figure 2.2 illustrates different levels in a network. R1 is a Level 2 router connected to the backbone. R2, R3 and R4 are Level 1-2 routers interconnecting

the backbone to the local areas. R5 to R10 are Level 1 routers connected to the different local areas.

2.3 Packets in IS-IS

There are three different types of IS-IS packets: *Hello, Link-State Packet (LSP), and Sequence Number Packet (SNP)*. All of the IS-IS packets consists of a header and a payload. The payload is made up by a number of variable length fields coded as *Type, Length, and Value (TLV)*. The first eight octets in the header is the same for all the IS-IS packets, but each packet type has some additional header fields, and of course the variable length payload.

Figure 2.3 shows the different fields in an IS-IS packet. Each of the eight first lines from the top consist of one octet. The additional header fields and the TLV fields are variable length.

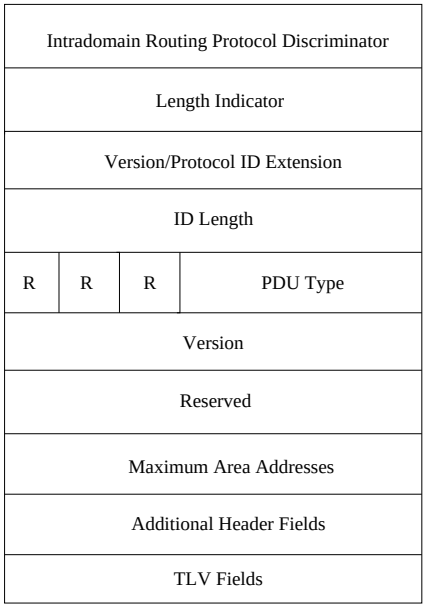

Figure 2.3: IS-IS packet

2.3.1 Hello Packet

The routers periodically send Hello packets every *Hello Interval* on all their IS-IS enabled interfaces. The *Hello Holdtime* is reset upon the reception of a Hello packet. If a router does not receive a subsequent Hello packet from a

given source within the Hello Holdtime, the adjacency is torn down. The Hello Holdtime is a multiple of the Hello Interval given by the *Hello Multiplier*. In this way the adjacency between neighboring routers is made and maintained.

There are different types of Hello packets for point-to-point (P2P) links and broadcast links. On broadcast links there are different PDU types dependent on the two circuit levels, but the format of the Hello packet is the same. Before a router can start to send or process received LSPs on an interface, the adjacency must be established.

The default value for the Hello Interval is 10 seconds, and for the Hello Multiplier is 3 times. This gives a Hello Holdtime of 30 seconds. On a broadcast link one of the router is elected as the "master" (see Chapter 2.3.1.1 on the following page for explanation), and this router has a Hello Interval of 3.3 seconds to early detect if it fails. The timers and parameters with their default values are listed in Table 2.1.

	Default value	
	Ordinary router	Psudonode (DIS)
Hello Interval	10 [s]	3.3 [s]
Hello Multiplier	3	3
Hello Holdtime	30 [s]	10 [s]

Table 2.1: Default values for Hello parameters.

The additional fields in a Hello packet are:

- **Circuit Type** — If the circuit is Level 1, Level 1-2, or Level 2.
- **Source ID** — The system identifier (SysID) of the originating router.
- **Holding Time** — The maximum time between two consecutive Hello packets before the router is considered no longer available.
- **PDU Length** — The length of the PDU, including the header.
- **Local Circuit ID** — An unique link identifier (P2P links).
- **Priority** — This value designates the priority to be the DIS on the LAN (broadcast links).
- **LAN ID** — The SysID of the DIS plus an one octet unique ID for this router assigned by the DIS (broadcast links).

Figure 2.4 shows the additional header fields in a Hello packet.

The length of the SysID must be consistent on all routers in the routing domain. If it differs, the Hello packet will be discarded at the receiver. Another requirement is that the maximum number of area addresses supported in a single router configuration must match between adjacent routers, unless the verifying

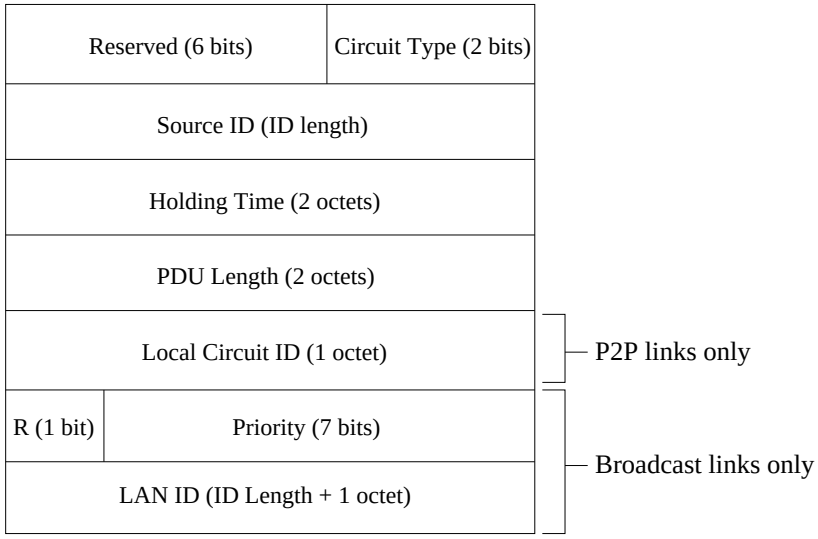

Figure 2.4: Additional header fields in a Hello packet.

router supports only a maximum value of 3. A router that receives a Hello packet with a different value , will discard the Hello packet.

Any changes to the links that generate changes to the TLV information compared to the last Hello packet, will trigger a immediate transmission of a new Hello packet.

2.3.1.1 Adjacencies on Broadcast Links

If the routers are installed to a broadcast link, e.g a LAN, one of the routers is elected as the *Designated Intermediate System (DIS)*. The router with the highest SNPA address (MAC address) gets the highest priority and is elected as the DIS. The DIS models the broadcast medium as a node, referred to as a *psudonode*. The psudonode minimizes the complexity of managing multiple adjacencies. There is no backup DIS, so the Hello Interval of the DIS is much shorter than the other routers to rapidly detect if the DIS fails. If the current DIS fails, a new DIS is elected. There are separate DISs for Level 1 and Level 2 routing. If a router with a higher priority than the current DIS is connected to a LAN, the new router will preempt the existing DIS without significant disruption of IS-IS operation. An election to or resignation from LAN DIS position will immediately trigger a Hello packet.

The routers connected to the LAN send Hello packets to well known broad-

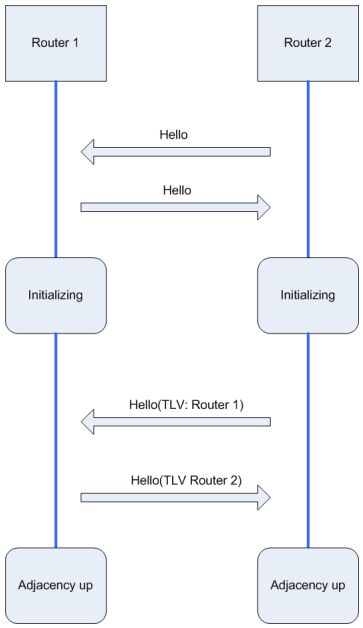

Figure 2.5: MSC for adjacency establishment on broadcast links

cast MAC addresses. The Hello packets include the MAC addresses to all neighbors the source router has received Hellos from. When a router receives a Hello packet it checks if it exist an adjacency with the transmitter in its adjacency database. If the neighbor is known, the holdtime is reset. If not, the receiver initialize an adjacency by waiting for subsequent Hello packets from the same transmitter. The two-way communication is confirmed and the adjacency is established when a Hello packet is received with the receiver's MAC address in the IS Neighbors TLV field. This method introduces reliable three-way handshake between two routers before the adjacency is established.

Figure 2.5 shows a message sequence chart (MSC) for the reliable establishment of adjacencies on broadcast links.

2.3.1.2 Adjacencies on Point-to-Point Links

On point-to-point links the adjacency is initialized by an ISH (Intermediate System Hello) through the ES-IS protocol. When a new point-to-point link is enabled, the router sends an ISH to the interface to see if it is a up and running router in the other end of the link. If it is, the receiving router checks if there already exists an adjacency with the transmitter by comparing the SysID in the ISH against its own adjacency database. The ISH is discarded if the SysID exists in the database. If not, the receiving router create a new adjacency and returns an IIH (IS-IS Hello) to the source. The state of the adjacency is set to "initializing" and the system type to "unknown". Then it waits for the source to

send a subsequent IIH. Upon receiving this IIH, the receiver moves the state of the adjacency to “up”, and changes the new neighbor’s system type to “IS”.

It is seen that this method does not give the reliable three-way handshake as in a broadcast link before the adjacency is established. The router that sends an subsequent IIH gets no confirmation whether the Hello packet reach the other end or not. This could result in a situation were one end of the link establish an adjacency, but its neighbor in the other end does not. Figure 2.6 shows a MSC for the establishment of an adjacency on a P2P link.

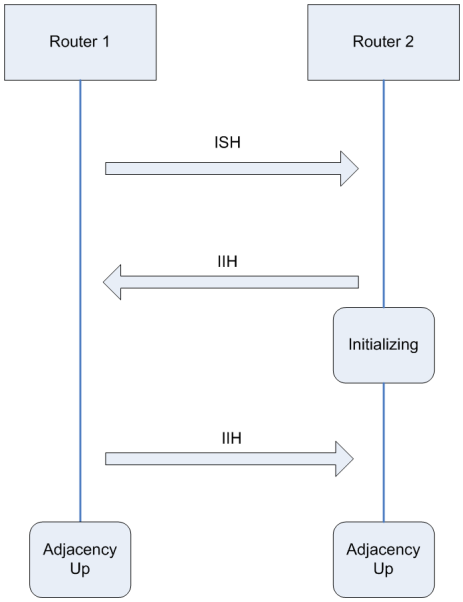

Figure 2.6: MSC for adjacency establishment on P2P links

Work has been done in IETF to standardize a more reliable method to establish the adjacencies on point-to-point links. By introducing a new TLV called “Point-to-Point Adjacency State TLV” (type 240), the adjacency between two routers is not established before a three-way handshake confirm bidirectional communication. This new method is made backward-compatible to the ISO 10589 procedure, making it possible for routers with and without TLV 240 to coexist in the same network. Another shortcoming in the original specification of IS-IS in ISO 10589 is the 8-bit Local Circuit ID field in the Hello header. This limit the number of defined links to 256 in a router. IETF is also working with this issue, to remove the limitation. The mentioned TLV 240 also includes an Extended Local Circuit ID field of 4 octets, which dramatically increases the number of possible links.

2.3.2 Link State Packets

Each IS-IS router builds a Link-State database (LSDB) that describes the topology of the area, and runs the SPF algorithm to obtain the best paths to all the destinations. A Link-State packet (LSP) contains routing information generated by an IS-IS router describing its immediate surrounding. After an adjacency is established, the routers periodically send LSPs to their neighbors. When a router receives a LSP, it installs the LSP in its Link-State database and sends a copy on all the other interfaces. In this manner the LSPs are exchanged through the network by the process known as *flooding* (see Chapter 3.4 on page 24). If the information in the LSP differs from the router's database, actions has to be carried out to synchronize the databases.

The information in a LSP includes:

- Area information.
- Adjacent routers.
- IP subnets.
- Metric information.
- Authentication information.

As mentioned are all IS-IS packets made up by an eight octets header, some additional header fields, and some variable length TLV fields. Figure 2.7 shows the additional header fields in a LSP packet.

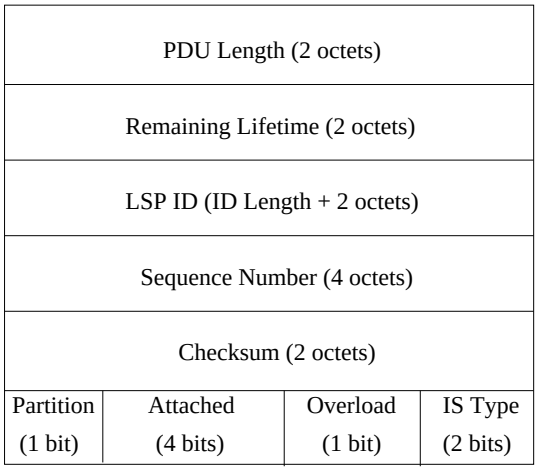

Figure 2.7: Additional header fields in a LSP packet.

The additional header fields for a LSP are:

- **PDU length** — Length of the entire PDU, including header and TLV fields.
- **Remaining liftetime** — Remaining time for the LSP to expire.
- **LSP identifier (LSP ID)** — The identification of the LSP. Distinguish LSPs from each other, and identify the source.
- **Sequence number** — The sequence number of the LSP.
- **Checksum** — Checksum of the LSP calculated over fields starting from the LSP ID.
- **Partition** — One bit flag. Set if source of LSP supports partition repair.
- **Attached** — Four bits flags to indicate attachment to another area.
- **Overload** — One bit flag. Set to indicate that the LSDB of the source is overloaded and that the processing and memory resources are limited. If set, the source is not used in calculating transit traffic, only if the source is the last hop.
- **IS type** — Two bits flag. Indicates if the originating router is a Level 1 or Level 2 router.

In the following subsections important timers and parameters for the LSDB update process are explained. A summary of the parameters and their default values are given in Table 2.2. The table shows also whether the parameter is specified in ISO 10589[16] or by Cisco[14], and if the feature is enabled in UNINETT.

2.3.2.1 Processing of LSPs

During convergence, before the network is stable, there could be a huge amount of changes in the LSDBs. A continuous update of the databases would be very bandwith- and processor load demanding, because of the exchange of all the LSPs. To limit this overhead to the network it is specified interval parameters for the calculation, generation, and (re)transmission of LSPs.

LSP transmission interval: Sets the minimum interval between consecutive transmissions of two LSPs. The default value is 33 milliseconds. This value is a result of the bandwidth and processor limitations 20 years ago. Compared to today's technology the default value is unnecessary high and contributes to long convergence time

	Exponential backoff			Single	Specified in ISO 10589/ enabled in UNINETT
	max-wait [s]	initial wait [ms]	second wait [ms]	Parameter	
SPF	10	5500	5500	—	no/yes
PRC	10	2000	5000	—	no/yes
LSP-generation interval	5	50	5000	—	no/yes
LSP-MaxAge	—	—	—	1200 [s]	yes/yes
LSP-refresh interval	—	—	—	900 [s]	no/yes
ZeroAgeLifetime	—	—	—	60 [s]	yes/yes
LSP-transmission interval	—	—	—	33 [ms]	no/yes
iSPF	—	—	—	120 [s]	no/no
Fast flooding	—	—	—	5	no/no

Table 2.2: Default values for LSDB update timers[14]

SPF calculation interval: When a router receives a LSP it waits a specific time before it runs the SPF process. This delay is known as the *SPF-initial-wait*, and has a default value of 5500 milliseconds. Periodic SPF runs are scheduled at increasing time apart until a threshold, and the interval becomes constant. The second SPF calculation is scheduled after a interval referred to as *SPF-second-wait*, and has a default value of 5500 milliseconds. Then the interval is doubled until it reach the threshold. This maximum interval is referred to as *SPF-max-wait*, and has a default value of 10 seconds. If the router does not receive a LSP within twice the time of the *SPF-max-wait*, the behavior is reset to initial wait time. This method to dynamically change the timer values is known as *exponential backoff*. It ensures rapid reaction to a topology change when the network is stable, and decreases the network overhead in case of flapping links.

PRC calculation interval: If a change in the network topology only influence leaf information, e.g. change in IP addresses, it is not necessary to run a complete SPF calculation. Instead a less processor intensive *partial route calculation* (PRC) can be run. As the SPF process, the PRC process has also a exponential backoff (*PRC-initial-wait, PRC-second-wait* and *PRC-max-wait*). The default values are 2000 ms, 5000 ms and 10 seconds respectively.

To speed up the convergence time it may seem tempting to minimize the initial delay for the SPF and PRC process. Caution should be taken before setting these timers too low. The SPF and PRC process preempt the other processes

in the router. This means that no PDUs are processed during the SPF/PRC calculation. It is important that the router floods at least the LSP that triggers the SPF/PRC calculation before it starts the calculation. If not the LSP will be delayed through the network which will give longer convergence time, as described in Chapter 2.3.2.2 on the facing page.

LSP remaining Lifetime

The remaining liftetime field in the LSP header includes a timer that tracks the age of the LSP. The reason for this timer is to purge the LSDB of stale information. The timer uses two parameters; *LSP maxage* and *LSP generation interval*.

LSP Maxage: The LSP maxage is the upper bound for the lifetime of a LSP. ISO 10589[16] specifies a default value of 1200 seconds (20 minutes) for the LSP maxage. This parameter is considered a protocol constant and must have the same value in all IS-IS routers in the network. When a LSP is generated the remaining lifetime field is set to the LSP maxage value before the LSP is flooded trough the network. The remaining lifetime decreases with time, and if it reaches zero, the LSP is purged out of the network. A router can initiate a purge of a corrupted LSP by setting the remaining lifetime of the LSP to zero, and reflooding the LSP to its neighbors.

If a LSP is not refreshed, the remaining lifetime field eventually reaches zero. All routers that have a copy of this LSP purge it from their LSDBs after a grace period. This grace period is referred to as *ZeroAgeLifetime* in ISO 10589[16], and has a default value of 60 seconds.

LSP Generation Interval: Ideally a router generates and floods a LSP as soon as possible without any delay. However it is specified a interval between the generation of two consecutive LSPs. This interval is known as LSP generation interval. The LSP generation interval is the frequency in seconds which the originating router regenerates a LSP. This happens no matter there has been changes to the network or not. When the LSP is regenerated the remaining lifetime field is reset to the LSP maxage value. The default value for the generation interval is 900 seconds (15 minutes), known as the *refresh interval*.

As for the SPF and PRC, the LSP generation interval has the same varieties to reduce bandwith consumption and processor load. The default values are 50 ms for the *LSP-initial-wait,* 5000 ms for the *LSP-second-wait*, and 5 second for the *LSP-max-wait*.

LSP Sequence Number

The first LSP from a router has a sequence number of 1. The sequence number field in the LSP header increase for every regenerating of the LSP. In this way it is easy to identify older LSP from newer versions, which helps the database synchronization process. If a router crashes and is reconnected to the network,

it will generate a new LSP with sequence number of 1. Given that the old LSP has not been purged yet, the other routers in the network has this old LSP with a higher sequence number in their databases. Upon reception of the new LSP with sequence number 1, one of the routers recognizes the recovered router by its LSP ID and sends a copy of the old LSP back to the source. In this way the recovered router can adjust the sequence number to be close to the value before it crashed.

2.3.2.2 Fast Flooding

The SPF process has priority over LSP flooding in the CPU[20]. This means that a router starts the SPF calculation before it floods the LSP to its neighbors, if both processes compete for the CPU capacity. Because of this is the arrival of the LSP delayed through the network. The SPF calculation is getting faster as the technology is improving, but this issue is a major contributor for the convergence time in large networks[11].

To solve this problem Cisco Systems has introduced a feature known as *fast flooding*[14]. If activated the command will tell the router to flood a certain number of LSPs before the SPF calculation is started. The default value is 5. By speeding up the flooding of LSPs, the fast flooding command helps to decrease the overall convergence time.

2.3.3 Sequence Number Packets

Sequence Number Packets (SNPs) are used in auxiliary mechanisms that ensures integrity of the LSP based routing information distribution process. When a router receives a LSP, it checks if the information in the LSP matches its own LSDB. If it does, the router just send a copy of the LSP to all the other interfaces, and no further actions is performed. If the LSP contains information unknown to the database, and vice versa, SNPs are used to update the missing information. This process is referred to as LSDB synchronization.

The SNPs consist of four key pieces of information extracted from the LSP header. This summary is necessary for a unique LSP identification.

A SNP contains:

- Link-State Packet Identifier (LSPID).
- Sequence number.
- Checksum.
- Remaining lifetime.

There are two different kind of SNPs; *Complete Sequence Number Packets (CSNP)* and *Partial Sequence Number Packets (PSNP)*. They share the same packet format and both carries a collection of LSP summaries. The basic difference between them is that a CSNP contains summaries of all known LSPs,

whereas a PSNP contains only a subset. CSNPs are used on both point-to-point links and on broadcast links to check consistency of the router's database.

PSNPs complement CSNPs in the synchronization process. Routers request transmission of current or missing LSP by using PSNPs on both point-to-point links and broadcast links, and on a point-to-point link PSNPs are used to acknowledge receptions of LSPs.

The following two subsections describe the LSDB synchronization process. The timers and their default values are listed in Table 2.3 on page 20.

2.3.3.1 LSDB synchronization on Point-to-Point Links

A CSNP is only sent once, when the adjacency has been initialized, on point-to-point links. The routers in both ends send a CSNP to the other router. Upon reception of the CSNP, the receiver compares this content with its own LSDB, and sends a PSNP back requesting the missing or outdated LSP summaries. The router in the other end respond by returning a LSP with the requested information. If one of the CSNP is lost in flight, the sender will notice the other end's missing information from the received CSNP. The router that received the CSNP then proactively sends copies of the missing LSPs. In this way the synchronization is achieved, without retransmission and further delay.

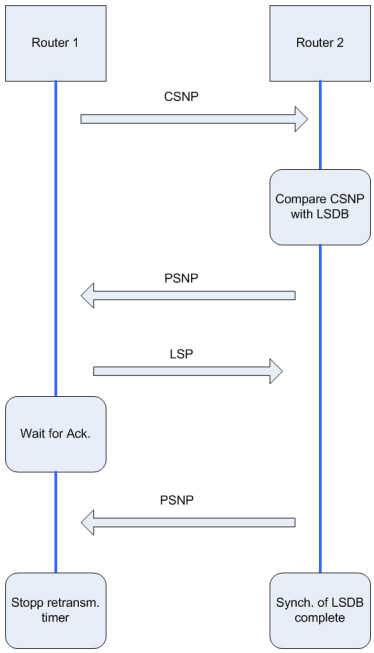

Figure 2.8: MSC for LSDB synchronization of P2P links

When a router receives a LSP, it returns a PSNP for acknowledgment. The router sending the LSP waits for the PSNP for a specified period, referred to as

the *retransmission-interval*, before the LSP is considered lost. This retransmission interval is set at a default value of 5 seconds. The router retransmits the LSP until a PSNP is received as acknowledgment. This makes the synchronization process on point-to-point links reliable.

Figure 2.8 on the facing page shows a MSC for the reliable exchange of SNPs to synchronize the LSDBs between neighbors.

2.3.3.2 LSDB synchronization on Broadcast Links

The DIS periodically multicasts CSNP on broadcast links. The default interval is 10 seconds, and is known as the *CSNP-interval*. If a router (re)connects to a broadcast network, it floods a LSP after an adjacency is established. At this moment the router has no information about the other routers in the network. The DIS receives this LSP and update its database. Then the DIS advertises a CSNP with summaries of all known LSPs. The new router receives this CSNP and return a PSNP requesting the missing information. Upon reception of the PSNP the DIS floods the requested LSPs, and when the new router receives these, it can synchronize its database.

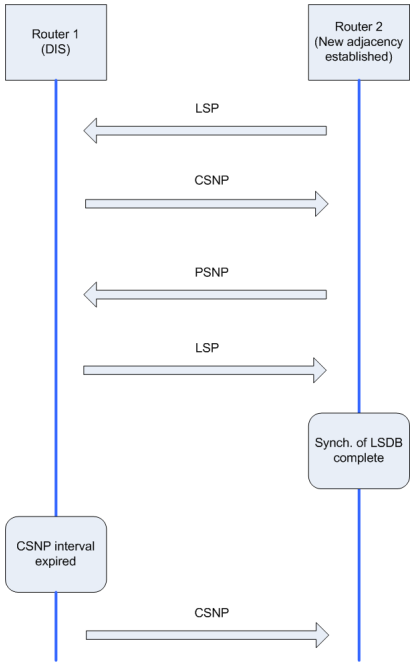

Figure 2.9: MSC for the LSDB synchronization of broadcast links

The database synchronization process on broadcast links is described as best effort. In contrast to point-to-point links, broadcast links do not employ acknowledge routines that guarantee reliable delivery of LSPs. Unreliable flooding requires a distinct mechanism to ensure database synchronization. This

mechanism is accomplished by periodic multicast of CSNPs on broadcast links. Periodic CSNP advertisement can be expensive with regard to bandwith consumption and processor load, but is a trade off for a simpler scheme to achieve reliable database synchronization.

Figure 2.9 on the previous page shows the exchange of SNPs for the LSDB synchronization process on a broadcast link. Compared to the P2P link shown in Figure 2.8 on page 18, it is seen that the synchronization process on the broadcast link is unreliable.

	Default value [s]
LSP-retransmit interval (P2P links only)	5
CSNP- interval (DIS only)	10

Table 2.3: Default values for LSDB synchronization timers.

Chapter 3

Fault handling

This chapter describes fault handling in networks with path restoration techniques, and the different contributors to the downtime during rerouting. UNINETT is a network of that kind, which make the theory in this chapter necessary to understand the methods and results for the case study and testing in Chapter 5. The chapter ends with a summary of the contribution of each component of the convergence time in UNINETT.

The text in this chapter is based on the paper "Achieving sub-second IGP convergence in large IP networks" [11] if nothing else is stated.

3.1 Why, and what is, Fault Handling?

The Internet was originally designed to carry "best-effort packets". This means the main goal was to provide connectivity through the network, at almost any delay cost. IS-IS adopted this thinking, which led to convergence time in order of tens of seconds[20]. Following the widespread deployment of real time applications such as VoIP and video streaming, much tighter *Service Level Agreements (SLAs)* are required. It is presented that sub-second IGP convergence is achievable without any compromise on stability.

Components of the convergence time:

1. Failure detection
2. LSP origination
3. Flooding
4. SPT computation
5. RIB and FIB update
6. Distribution delay

The total convergence time is the sum of this steps. These different components are described in the following sections.

Translation to common use in literature

In the problem description the convergence time is referred to as only a part of the downtime in case of a rerouting. Common use in literature describes the whole downtime as the convergence time. Apart from the problem description, the rest of this report adapt the common use of the convergence expression. This means the three contributors listed in the problem description is "converted" to the six components described in this chapter.

Contributors converted to Components:

- Time to detect failure: Start of component 1.
- Delay before failure is considered permanent: End of component 1, and partly component 2 and 4.
- Convergence time: Rest of component 2 and 4, and component 3, 5 and 6.

3.2 Failure Detection

The faster a failure is detected, the faster the network can adapt to this failure and minimize the packet loss.

In a network were the data link hardware comes into play, such as SDH alarms, the detection times can be very short (<100ms)[21]. However, there are media that do not support this feature, e.g. Ethernet, and some media may not detect certain kinds of failures, e.g. failing interfaces or forwarding engine components. In these cases it is the routing protocol that detect failures by exchanging hello PDUs between neighbors. The hello protocol in IS-IS is described in Chapter 2.3.1 on page 8. The default values gives a maximum detection time of 30 seconds which is far too much for most applications. It is possible to change the parameters for the hello PDUs and timers to reduce the detection time. This method can give detection times in order of sub-second, but implies high overhead to the network.

UNINETT consists of both SDH- and Ethernet P2P links, and Ethernet broadcast links. Ethernet links may be separated by switches, both P2P- and broadcast links. In a switched Ethernet the failure detection depends on the hello protocol, which implies high delay or high overhead.

3.2.1 Bidirectional Forward Detection

Within IETF work is in progress to specify a method to detect faults in the bidirectional paths between two forwarding engines[22]. This method is called *Bidirectional forward detection (BFD)* and is a simple "Hello protocol" intended to

rapidly detect link or node failures. BFD can be utilized by network components for which their integral aliveness mechanism are either too slow, inappropriate, or nonexistent.

Figure 3.1 illustrates the problem on a switched Ethernet connection. The failure between switch A and switch B is invisible to router A and router B because both routers detect a carrier on their interfaces, unaware of the separation by the switches. The failure detection depends on the hello holdtime in IS-IS. If BFD is implemented between router A and router B the failure detection would be faster. Because it is possible to route the traffic via router C, the downtime experienced by user A and user B would be decreased.

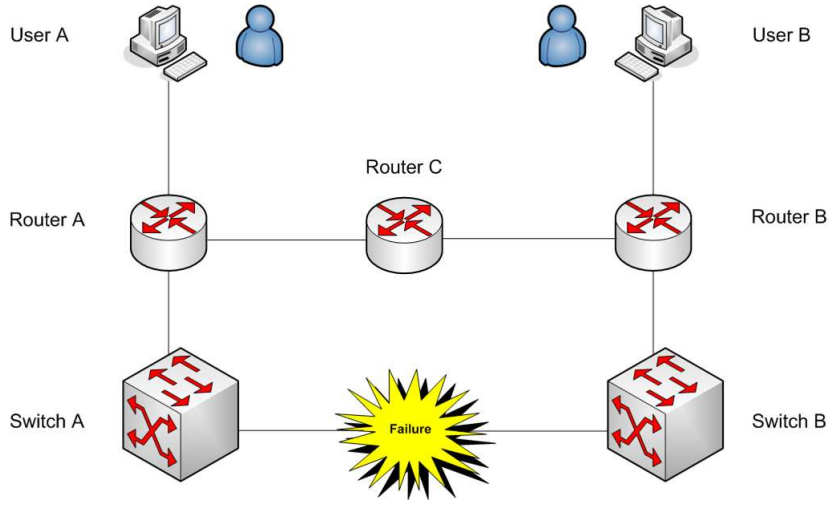

Figure 3.1: Failure to a switched Ethernet

The purpose of BFD is to verify connectivity between a pair of systems for a particular data protocol across a path. The path may be of any technology, length, or OSI layer[23].

The main goal of BFD is to provide low overhead, short duration detection of failures in the path between adjacent forwarding engines. The path includes the interfaces, data links, and to the extent possible; the forwarding engines themselves. An additional goal is to provide a single mechanism that can be used for aliveness detection over any media and at any protocol layer. Different medias and protocol layers may have a wide range of detection times and overhead, which implies different detection methods. With BFD as the common mechanism, it will avoid a proliferation of different methods.

BFD operates on top of any data protocol being forwarded between two systems. It is always run in a unicast, P2P mode. A separate BFD session is created for each communication path and data protocol in use between two systems. The BFD packets is transmitted in UDP packets within IP packets across single hops of IPv4 or IPv6[24].

BFD can also be useful on arbitrary paths between systems[25]. These paths may be unpredictable and span multiple network hops. Furthermore, a pair of systems may have multiple overlapping paths between them.

Each system estimates how quickly it can send and receive BFD packets. This follows an agreement with its neighbors about how rapidly detection of failures will take place. These estimates and agreements can be modified in real time in order to adapt to unusual situations. On a broadcast link this design also allows for fast and slow systems to participate with different speed.

3.3 LSP Origination

An important issue to achieve fast convergence is a rapid dissemination of updated LSPs. When there is a topology change it is of interest to inform all nodes in the area about this change as fast as possible. This approach could lead to a massive overhead to the network and CPU load in case of flapping links.

To adapt to this problem dynamic LSP timers has been introduced in stead of traditional static timers[13]. The method with dynamic timers is called exponential backoff and is described in 2.3.2.1 on page 14.

3.4 Flooding

Flooding is the process of distributing the LSPs to all nodes in the network. The total flooding time is the sum at each hop of the bufferisation, serialisation, propagation and the IS-IS processing time:

Bufferisation is the time a packet is queued in input and output buffers before it is processed. Most ISPs are capacity planned outside congestion and routers prioritise routing updates through input and output buffers.

Serialisation is the time it takes to clock a packet on the link. A typical data link layer frame of 1500 bytes are sent in less than $5\mu s$ if the link speed is 2.5 Gbps.

Propagation is the time it takes to send the packet between the routers. The law of physics limits this delay by the speed of light which is 300 km/ms in vacuum. (A bit slower in a fiber optic cable.)

IS-IS processing time is the time the router takes to process the packet for flooding. From the packet arrives from the input buffer, until it is sent to the output buffers. By default the LSP is flooded after the SPF calculation, but this can be altered with the fast flooding command described in Chapter 2.3.2.2 on page 17. The IS-IS processing time is also related to the LSP-transmission timer described in Chapter 2.3.2.1 on page 14.

The bufferisation, serialisation, and propagation delay is negligible with today's technology. The IS-IS processing time has also very little affect to the total

convergence time if fast flooding is enabled, and the LSP-transmission timer is set to a minimum value.[11]

3.5 SPT Computation

In a network with IS-IS as the IGP, every router calculate a shortest path tree (SPT) for the whole network. This SPT calculation takes a lot of CPU resources, and stalls all other processes during the computation. Because of this it is important that the SPT algorithm is not run to often in case of flapping links, and to flood at least the LSP that triggered the calculation before the SPT is run.

The first issue is controlled in the same manner as LSP origination, with dynamic timers called exponential backoff explained in Chapter 2.3.2.1 on page 14. The latter is achieved when fast flooding is enabled as mentioned in Chapter 3.4 on the facing page. These timers have as default an high initial wait value, which is a major contributor to the overall convergence time.

In many cases of a topology change, the change does influence only parts of the network. Then it is not necessary to run a complete SPF calculation. *Incremental SPF (iSPF)* is a algorithmic optimisation which decrease the computation time. If the iSPF command is enabled, the router needs a "warm up" period before activating the iSPF algorithm after a (re)boot. The default value is 120 seconds. See Table 2.2 on page 15 for a summary of the LSP timers and parameters used in the SPT computation.

3.6 RIB and FIB Update

Every router has a *routing information base* (RIB) and a *forwarding information base* (FIB) that need to be updated in case of a topology change. This update process duration is showed to be linearly dependent with the number of modified prefixes, with the main CPU performance as the key bottleneck in the convergence process[11].

Three approaches exist to minimise the update component:

1. Network design rule to minimise the number of IGP prefixes.

2. Protocol and implementation optimisation to allow prioritisation of some prefixes during update.

3. Intrinsic optimisation of the table management code.

For a network with the size as UNINETT (<100 prefixes) the RIB/FIB update process is fast (<<300 ms), and has little influence to the overall convergence time.

3.7 Distribution Delay

When the main CPU in the router has updated its RIB, it has to distribute this information to the CPUs on the linecards. This imposes a delay in the convergence process in the order of 50-70 ms [11].

The distribution process can be optimised with the use of multicast transport between the main CPU and the CPUs on the linecards, but in UNINETT this delay is a small contributor to the overall convergence time.

3.8 Summary of Convergence Components

Figure 3.2 shows a simple sequence chart for the fault handling. The most important contributors to each component of the convergence time are pointed out in the figure.

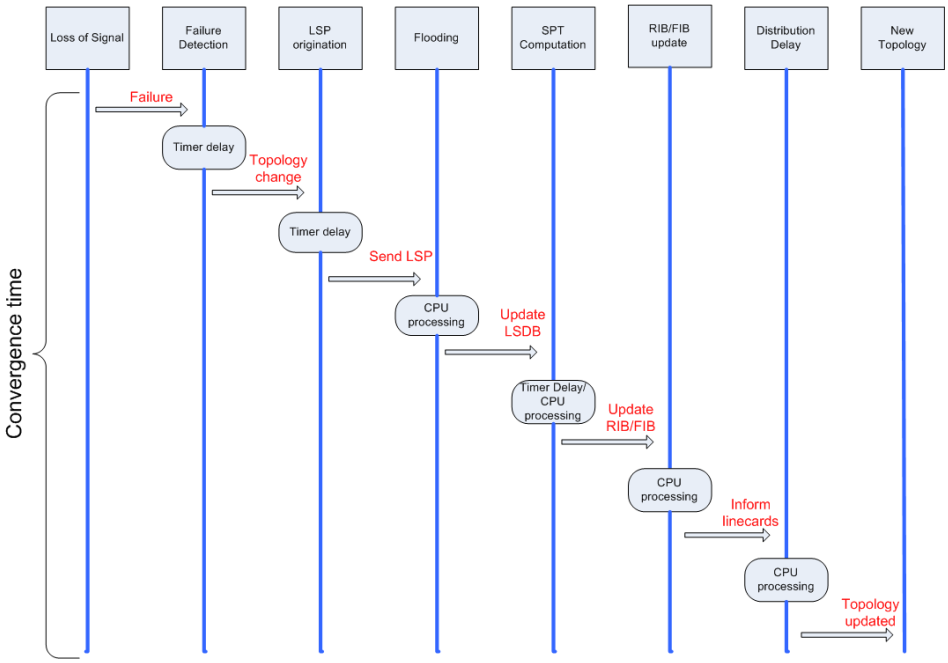

Figure 3.2: Sequence chart of the components to the convergence time

The SPT computation component is the most significant in UNINETT today, due to the high initial wait timer. Other significant contributors in UNINETT are the failure detection and the flooding components as discussed in Chapter 5.2.

Chapter 4

Micro Loops

This chapter discuss the problem with micro loops, and describes different strategies and methods to solve the problem. Micro loops implies periods with packet loss, which delay the end-to-end characteristics and reduce the overall QoS. Section 4.5 on page 33 gives a brief summary of the different strategies, and discuss the pros and cons for their practicable implementation in UNINETT.

The methods described in this chapter are not used in either the case study nor the test in Chapter 5, but is the background for some of the conclusions in Chapter 6.

4.1 An Introduction to Micro Loops

It is a well known problem that transient loops occur during topology changes in LS routing protocols. These incidents are referred to as micro loops and happens because the FIBs in the affected routers is not updated simultaneously. If the topology of the network changes, the FIBs in the affected routers need to be updated. During this convergence phase micro loops may come into existence. A change in the topology can be triggered by a sudden failure or by a predictable maintenance operation, and happens both when the node or link goes down and comes up again. In all these cases micro loops are a problem. Just a change in the cost metric on a link can generate micro loops, even without any loss of connectivity.

Figure 4.1 illustrates the problem with micro loops. The shortest path between user B and user A is R1 to R4 with a cost metric 1. If the link between R1 and R4 fails, the new shortest path is R1 via R2 and R3 to R4. R1 is the first node that detects the failure and updates its LSDB. R1 reroute the traffic to R2. R2 sends the traffic back to R1 because seen from R2 the shortest path to user A is via R1 and R4. The traffic will loop between R1 and R2 until R2 has updated its LSDB. Then R2 reroute the traffic to R3, but the same loop happens here; the traffic will loop between R2 and R3 until R3 has updated its LSDB.

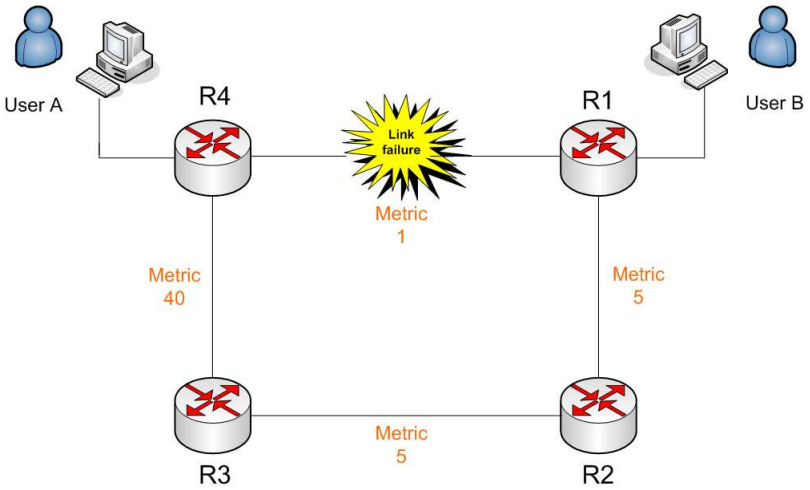

Figure 4.1: Micro loop

When the link between R1 and R2 comes up again, it could also be a problem with micro loops. If R3 updates its LSDB before R2, which in turn updates its LSDB before R1, the same loops are constructed. Micro loops impose packet loss, which decrease the overall QoS experienced by the users.

There are three basic classes of micro loop control strategies[10]:

1. Micro loop mitigation
2. Micro loop prevention
3. Micro loop suppression

These three different classes are described in the following sections. The text is based on the IETF draft "A Framework for Loop-free Convergence" [10] if nothing else is stated.

4.2 Micro Loop mitigation

The micro loop mitigation strategy is to reduce, but not eliminate, the impact of the micro loops. Such a scheme can not guarantee the QoS during the convergence phase after a topology change.

Path locking with safe-neighbors (PLSN) is the only known mitigation strategy [9]. It works by defining a "next-hop safety condition" that the routers have to satisfy before they update their FIBs in case of a topology change. The safety condition defines when it is safe for a router to switch to a different neighbor as its next-hop for a specific destination.

Safety condition criteria in a network with symmetric link costs:

- The router considers the new next-hop neighbor as its loop-free neighbor based on the topology before the change.

- The router considers the new next-hop neighbor as its downstream neighbor based on the topology after the change.

The first requirement ensures that the neighbor has not been forwarding traffic for the specific destination to the source router before the topology change. The second requirement makes sure that the neighbor does not forward traffic for the specific destination back to the source router after the topology change.

The difference in the criteria before and after the topology change is to make sure that the source router and its new next-hop neighbor do not recursively consider each other as safe next-hops when they learn about the change.

Safety condition criteria in a network with asymmetric link costs:

- The router considers the new next-hop neighbor as its downstream neighbor based on the topology both before and after the change.

Based on these criteria the destinations are classified by each router in three different classes:

1. Type A destinations — Destinations unaffected by the topology change, and destinations whose next-hop satisfy the safety condition after the topology change.

2. Type B destinations — Destinations whose primary next-hop do not satisfy the safety condition, but whose can be sent via another next-hop that satisfy the safety condition.

3. Type C destinations — All other destinations.

After a topology change the Type A destinations are immediately changed to go via the new topology. Type B destinations are immediately changed to go via the next-hop that satisfies the safety condition. Type C destinations are changed when all routers have changed their Type A and B destinations. When all routers have changed their Type C destinations it is safe to change the Type B destinations to the primary next-hop.

The mitigation strategy produces a significant reduction in the number of links that are subject to micro looping. However it is only a partial solution, because a link between a pair of type C routers would be vulnerable to micro loops.

4.3 Micro Loop prevention

The micro loop prevention strategy controls the convergence of a network when the topology changes in such a way that no micro loops form. Such a scheme prevents the collateral damage that occurs to other traffic for the duration of the micro loops. If this strategy is combined with any fast repair method, the network will still be available during the convergence time.

Eight micro loop prevention methods have been proposed[10]:

1. Incremental cost advertisement
2. Nearside tunneling
3. Farside tunneling
4. Distributed tunnels
5. Packet marking
6. New MPLS labels
7. Ordered FIB update
8. Synchronized FIB update

These methods are shortly described in the following subsections.

4.3.1 Incremental Cost Advertisement

Normally the cost metric is changed in one step from its assigned value to infinity when a link fails, and back to its initial value in one step when the failure is repaired. The incremental cost advertisement method change the cost metric in small steps in case of a link failure or repair.

This approach has the advantage that it need no change to the routing protocol. It will work in any network that uses LS IGP because it needs no cooperation from the other routers. The disadvantage is the time it takes to converge. In a large network with high cost metrics this method can be extremely slow.

4.3.2 Nearside Tunneling

The nearside tunneling method creates a virtual network using tunnels in case of a topology change. The tunnels are established within the old topology, and carries the traffic that is affected by the topology change. Each router builds a tunnel to the closest (nearside) router adjacent to the failure.

When all the traffic are in the virtual network, the real network is allowed to converge on the new topology. No micro loops will form because the affected traffic is carried in the virtual network. When the network has converged, the tunnels are removed and the traffic is sent through the new topology.

This approach is much faster then the incremental cost advertisement method. The disadvantage is that it requires all the routers in the network to be capable of high performance IP tunneling.

4.3.3 Farside Tunneling

The farside tunneling method operate in the same manner as the nearside tunneling method, by creating tunnels to carry the affected traffic in case of a topology change. The difference is that the farside tunneling method creates tunnels terminating at the farside of the failure.

The advantage of this method is that it gives a more uniform distribution of the repair traffic than is achieved using the nearside tunneling method. And in case of a node failure, the decapsulation load on any router is reduced.

4.3.4 Distributed Tunnels

The distributed tunnel mode of operation is similar to the nearside- and farside tunneling methods. Each router calculates its own repair and forwards traffic affected by the failure using that repair. The objective is to get a more optimal routing than the farside tunneling method where the traffic is forced to go to the farside of the failure.

4.3.5 Packet Marking

This method implies some sort of marking of the packets to differentiate them to one of: the old topology, a transition topology, or the new topology. The marking could be achieved by the *type of service (ToS)* bits in the IP header[26].

There are three problems with this solution:

1. The packet marking bit may not be available.

2. It introduces a non-standard forwarding procedure.

3. It would increase the size of the FIB.

4.3.6 MPLS New Labels

This method for a MPLS network is similar to the mentioned methods using tunnels. In case of a topology change the routers check if they need to change their FIBs. If the FIB has to be updated and the old path traverse the failure, the router issues a new label to its neighbors. The new label is used to lock the path during the transition.

4.3.7 Ordered FIB Update

With the ordered FIB (oFIB)[27] mechanism, micro loops are prevented by correctly sequencing the FIB updates in the routers. This mechanism can be used in the case of non-urgent link or node shutdowns and restarts, or link metric changes. It can also be used in the case of a sudden link or node failure, but this implies that a complete repair path is provided to the destinations. The repair path makes it possible to convert the failure into a non-urgent topology change.

In case of a non-urgent topology change, each router calculate a rank that decides the time at which it can safely update its FIB. To avoid micro loops, a router must not update its FIB until all other routers that send traffic via this router have first updated their FIBs.

To speed up this loop-free convergence process, completion messages are exchanged between the affected routers. When a router has received a completion message from all the neighbors in its *waiting list*, it is allowed to update its FIB even if the ranking timers has not expired. When the FIB is updated, the router sends a completion message to all the neighbors in the *notification list*, which are waiting for it to complete.

This ordered update process of the FIBs has to be accomplished before the state of the affected link or node is changed.

4.3.8 Synchronized FIB Update

As mentioned in the start of this chapter, the inconsistency in the FIBs is the root of the micro loop problem. If the update of the FIBs in all of the routers is synchronized there would be no micro loops. One approach is to build two FIBs in each router. A FIB for the old topology, and another for the new topology. When all routers have updated the new topology FIB, they can switch to this FIB in a synchronized manner.

This method has some issues. The need of two FIBs in each router introduces a scaling problem, and the synchronization process gives high requirements to the system clocks. Work is in progress within IETF to define a mechanism that enables routers to agree on a common convergence delay time[28]. Such a mechanism will synchronize the FIB update and eliminate micro loops.

4.4 Micro Loop suppression

This strategy attempts to eliminate the collateral damage done by micro loops to other traffic. It makes no attempt to prevent the micro loops, but if a loop is detected, the task is to protect the other traffic by suppressing the looping packets. This could be achieved by a packet monitor that detect if a packet is looping, and drops it.

The advantage with this strategy compared to the mitigation and prevention strategies is that it does not extend the convergence time. The disadvantage is

that no effort is made to forward the looping packets to its destination, which implies packet loss for those connections.

4.5 Comparison of Micro Loops Strategies

There are many different strategies and methods to deal with the micro loop problem. The question is how much packet loss the micro loops introduce in the network, and at which cost the problem is going to be solved. The different techniques impose different amount of delay and overhead to the network.

Micro loops are a small contributor to the overall convergence time in UNINETT today as seen in Chapter 5.3. The continuous increase in the requirements for QoS make all contributors to downtime a possible improvement. Because of that is it of interest to study the possibility to implement a method to reduce the downtimes due to micro loops in UNINETT.

The mitigation strategy: Reduces the number of micro loops, but does not remove all possibilities of a loop to happen. The PLSN method introduce some overhead to the CPU, and some delay to the convergence phase. The link loads are increasing due to looping packets. Because of that is the mitigation strategy and PLSN not the recommended technique for dealing with the micro loop problem in UNINETT.

The prevention strategy: Eliminate all formation of micro loops. The different methods impose different impact to the CPU- and link overhead, and to the delay of the convergence time. The oFIB method[27] introduce some overhead to the CPU because of the rank calculation and processing, and some extra overhead to the network due to the exchange of completion messages. The convergence time is delayed during this process. The overhead is small in the big picture, and the completion messages contribute to speed up the convergence phase. This makes the oFIB method a good candidate to solve the micro loop problem in UNINETT!

The suppression strategy: No delay in the convergence time, and no disturbance to the non looping traffic. Imposes some overhead to the CPU because of the detection of looping packets. These issues make this strategy seem better than the two preceding. The drawback with this strategy is the packet loss due to the discard of looping packets. Because of that is the suppression strategy not recommended for implementation in UNINETT.

The oFIB method[27] is a good candidate to eliminate micro loops in UNINETT!

Chapter 5

Measurements and analysis

This chapter describes the different measurements performed in UNINETT, and a test lab established to document the pros and cons when tuning the IS-IS parameters in routers.

The chapter ends with a comparison of the observation from a case study in UNINETT and the results from the test lab.

5.1 Measurements in UNINETT

As mentioned in the problem description are UNINETT measuring packet loss in connection with link and router failures. There are established 14 measuring probes at different locations in UNINETT. The probes makes it possible to observe periods with packet loss. It is of interest to compare these observations with the LSPs and the logs in the routers, to analyse the different components to the convergence time. In addition is the graphical load map of the links in UNINETT a useful tool for the study.

By comparing the time stamps from the probes with the LSP collection and the router logs, it is possible to analyse the different contributors to the packet loss during convergence. The contributors are described in details in Chapter 3, and the method for analysis is described bellow in Section 5.2.

5.1.1 Measuring Probes

Each probe are sending a non-stop stream of PDUs to a subset of the other probes. The granularity between two consecutive PDUs is usually 500 ms. (Some connections have a 100 ms granularity.) The PDUs is tagged with an incremental sequence number. By looking at the sequence number, the receiver discovers if there are PDUs missing, and reports a period of packet loss in its log. If the receiver discover one or more missing PDUs, it log the difference in time between the last received PDU before the loss started and the first received PDU

after the loss stopped. Due to the mentioned granularity, the smallest possible observed period of packet loss is 1 s (or 200 ms).

Work is in progress in UNINETT to make a web presentation of the observations from the measurement probes. This service is at the time of writing under construction, but the current version is available at UNINETT's web[29].

5.1.2 LSP Collection

All LSPs flooded in UNINETT's backbone are collected by a server connected to Trd-gw in Trondheim. This collection is available at UNINETT's web[30].

The server logs:

- LSP number.
- Time stamp when it receives the LSP.
- Sequence number.
- Lifetime stamp.
- TLV type.
- TLV value.

This LSP information alone is not enough as explained below. But when compared with the router's logs and the data from the measurement probes, analysis concerning the components of the convergence time can be performed.

The log is saved "on the fly" and updates the history for the present date. The log is reset every midnight, making the LSP number starting at 1.

It is worth mentioned that the time stamp says when the server receives the LSP, **not** when the LSP is generated by the source! The LSP is delayed through the network, but how much depends on the queues and CPU loads in the path from the source to the server.

Another inaccuracy is the lifetime stamp; it is decremented individually in each router in the path. This makes it impossible to calculate when the LSP was generated by looking at the lifetime stamp.

5.1.3 Router Log

The routers log when the IS-IS timers are triggered, and what triggered them. This includes the SPT computation and LSP generation. Another interesting information is the data-link layer triggers. In case of a link failure the data-link layer (SDH or Ethernet) detects the loss of signal, and after the expiration of a timer, the network layer (IS-IS) is informed.

The drawback with the router log is that the history is very short and is not saved. Because of that is it not possible to look up older information than a few hours back. If there is an interesting incident, the log has to be read before it

is too late. Another, and more important drawback, is that the IS-IS logs are written with 1 s interval. This makes it impossible to compare information in the sub-second range, and the information from the SPF- and LSP log is more or less useless.

5.1.4 Graphical Load Map

In UNINETT there is collected information about the link loads. This information is presented in a graphical map available at UNINETT's web[31]. Figure 5.1 shows the graphical map of UNINETT on the 7Th of May 2007. By clicking on the links it is possible to navigate in the history log to find the capacity, link load, packet discards, etc. on each link at different time scales. Information like this is useful when studying the impact of a topology change regarding rerouting paths.

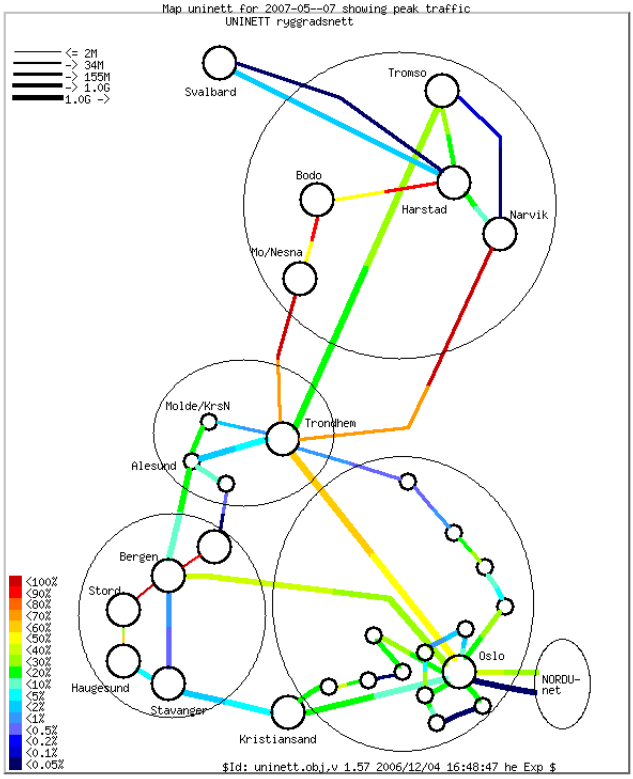

Figure 5.1: Graphical load map[31] of UNINETT's backbone the 7Th of May 2007

5.2 Contributors to the Convergence Time

The mentioned measurement probes in UNINETT show typically about 6-9 seconds period of packet loss in case of a topology change, as will be discussed in Chapter 5.3. The period of packet loss depends on the convergence time; the time it takes for the nodes in the network to adapt to the changes.

Chapter 3 lists six components of the convergence time. This section describes the analysis process to divide the convergence time in this different components, their margin of error, and their influence on the packet loss in UNINETT. The two last components are analysed together because the available measurements make them impossible to separate, and their total contribution to the convergence time is small in the big picture. An example of the analysis follows in Section 5.3.

5.2.1 Failure Detection

Compare the start of loss from measurement probes with the data-link layer trigger. The failure detection delay differs a lot from SDH and non-switched Ethernet, to switched Ethernet:

Failure detection in SDH- and non-switched Ethernet networks is fast ($<$ 100 ms), but the granularity from the measurement probes are too slow to state the exactly time of failure. Because of that is it impossible to observe this component, but after all it is not a big contributor to the convergence time.

Switched Ethernet connections does not have an end-to-end failure detection mechanism and depends on the expiration of the IS-IS hello holdtime timer. This timer has a default value of 30 s which makes the failure detection component of switched Ethernets a major contributor to the convergence time and packet loss! Fortunately UNINETT consists of few switched Ethernets and the failure detection component is not a big contributor to the overall convergence time.

5.2.2 LSP Origination

Compare the data-link layer trigger with the LSP generation trigger in the router log. CPU load may delay the logging, but the LSP generation trigger has a delay timer configured by the exponential backoff algorithm. This timer alone is more or less the LSP origination component of the convergence time. In UNINETT the LSP origination is a small contributor to the overall convergence time and thereby the packet loss.

5.2.3 Flooding

Compare the LSP generation in the router log with the time stamp in the LSP collection. The delay in the path from the source to the collecting server may differ due to change in CPU load and queue length.

If fast flooding is not enabled and small values for the delay of the SPT computation is chosen, the flooding process may be further delayed. This is

because the SPT process preempt the LSP processing, and delays the flooding of the LSPs.

The pacing timer controlled by the LSP-transmission parameter could also delay the flooding process in case of a failure generating multiple LSPs in the network.

If the flooding is delayed, the synchronization of the LSDB is also delayed. This may lead to routing loops and packet loss. However, the flooding component is not a big contributor to the convergence time and the packet loss in UNINETT.

5.2.4 SPT Computation

Compare the SPF- and LSP log in the nearside router to the failure. If fast flooding is enabled and minimum values for the initial delays are chosen, the SPT calculation should start right after the LSP is generated. Because of the 1 s interval from the router log, it is not possible to observe the short time the SPT computation takes.

According to [13] a full SPF calculation in a network with the size and CPU speed like UNINETT, should take approximately 50 ms. It is seen that the SPT computation itself is a small contributor to the convergence time. However, the SPF and PRC parameters have configured delay timers between the triggering and the calculation.

In UNINETT the default values for these parameters are chosen which impose a period of many seconds of packet loss, and make them major contributors to the convergence time!

5.2.5 RIB and FIB update/Distribution delay

Compare the SPF log from the nearside router to the failure with the end of packet loss from the measuring probe, given the downstream neighbor does not perform any routing loops. Because of the mentioned SPF log interval it is impossible to observe the small delays these components introduce.

According to [11] the RIB and FIB update in a network like UNINETT should take less than 300 ms, and the distribution delay should be less than 70 ms. The comparison of the network is based on the number of prefixes and the CPU speed in the routers.

Therefore it is assumed that these components are small contributors to the overall convergence time in UNINETT.

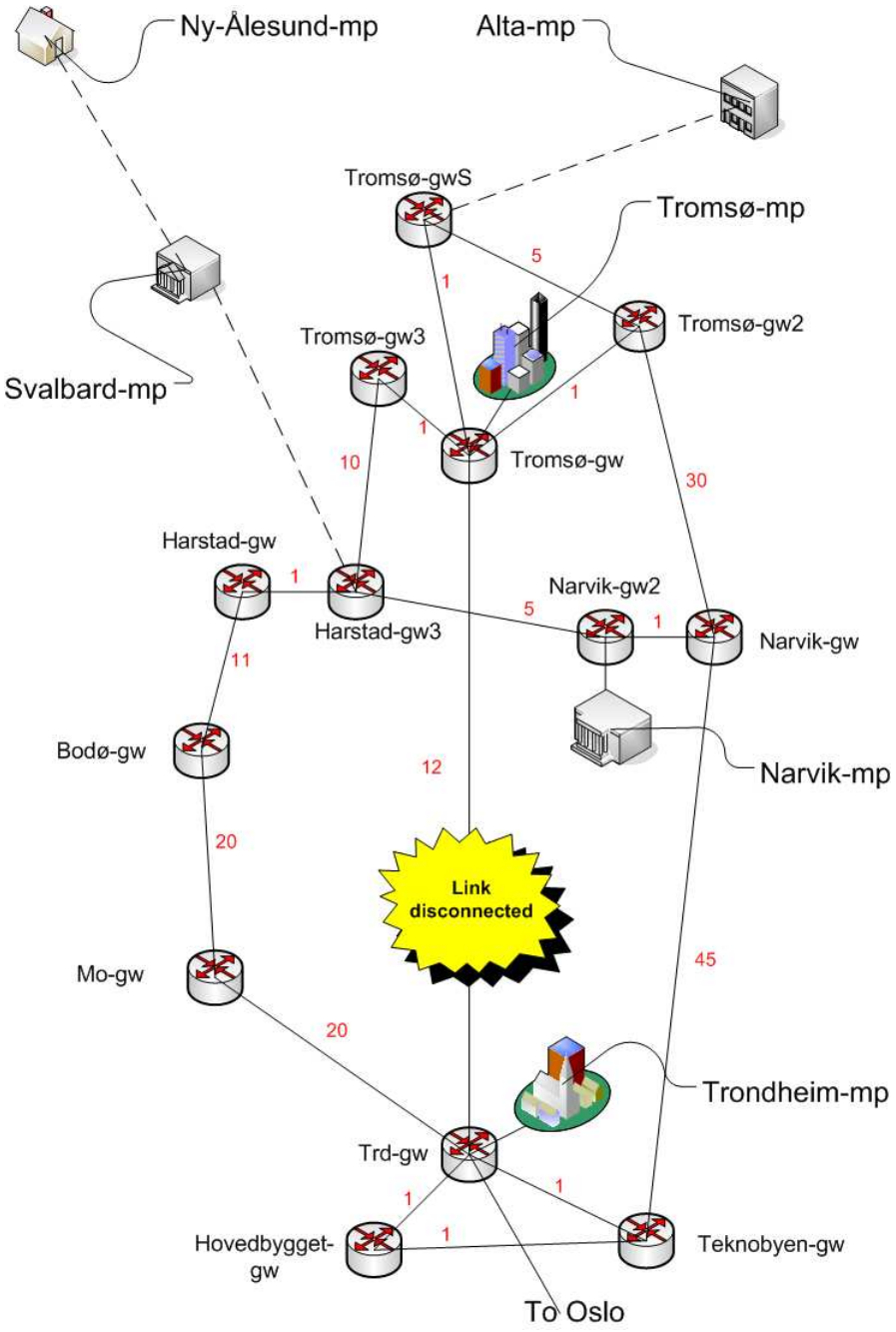

Figure 5.2: Topology of UNINETT in Northern-Norway

5.3 Case Study

In the morning of the 7Th of May 2007 it was performed maintenance work to the 2.5 Gbps link between Trondheim and Tromsø. The link was disconnected for a few hours, and the traffic was rerouted via Narvik or Bodø.

5.3.1 The Topology of Northern Norway

Figure 5.2 shows the topology of UNINETT's Level 2 backbone in Northern Norway. The dotted lines to Svalbard, Ny-Ålesund and Alta illustrates Level 1 areas. The measuring probes are labeled with the location name and -mp.

The cost metrics to each link are written in red. The shortest path between to nodes are found by sum up the metrics on the path. All measurement probes in Northern Norway have their shortest path to Trondheim and Oslo via the affected link. After the failure the shortest path between Tromsø and Trondheim is: Tromsø-gw–Tromsø-gw3–Harstad-gw3–Narvik-gw2–Narvik-gw–Teknobyen-gw–Trd-gw, or Tromsø-gw–Tromsø-gw3–Harstad-gw3–Harstad-gw–Bodø-gw–Mo-gw–Trd-gw. This phenomena is called *load balancing,* and is due to the same cost metrics on the paths from Harstad-gw3 to Trd-gw via Narvik or Bodø. The load balancing decides the path depended on the source and destination addresses. In this way the traffic from a given source to a given destination always follows the same path.

5.3.2 Triggers in Router Logs

The router logs from Trd-gw and Tromsø-gw on the 7Th of May 2007 are shown in Table 5.1 and 5.2. It is seen that the two routers reacted to the "*bad news*" just 11 ms in difference, but reacted to the "*good news*" with much more difference.

Time	Trigger
01:10:34.983	LINK-3-UPDOWN: Interface POS6/0, changed state to down
01:10:35.619	SONET-4-ALARM: POS6/0: SLOS
01:10:35.983	LINEPROTO-5-UPDOWN: Line protocol on Interface POS6/0, changed state to down
06:23:07.177	SONET-4-ALARM: POS6/0: B1 BER exceeds threshold, TC alarm declared. SONET-4-ALARM: POS6/0: B2 BER exceeds threshold, TC alarm declared.
06:23:13.177	SONET-4-ALARM: POS6/0: SLOS cleared
06:23:17.177	SONET-4-ALARM: POS6/0: B1 BER below threshold, TC alarm cleared. SONET-4-ALARM: POS6/0: B2 BER below threshold, TC alarm cleared. SONET-4-ALARM: POS6/0: SD BER below threshold, alarm cleared. SONET-4-ALARM: POS6/0: SF BER below threshold, alarm cleared.
06:23:17.277	LINK-3-UPDOWN: Interface POS6/0, changed state to up
06:23:18.277	LINEPROTO-5-UPDOWN: Line protocol on Interface POS6/0, changed state to up
06:23:25.689	CLNS-5-ADJCHANGE: ISIS: Adjacency to tromso-gw (POS6/0) Up, new adjacency

Table 5.1: Router log from Trd-gw on the 7Th of May 2007

Time	Trigger
01:10:34.994	LINK-3-UPDOWN: Interface POS1/1, changed state to down
01:10:35.330	SONET-4-ALARM: POS1/1: SLOS
01:10:36.870	LINEPROTO-5-UPDOWN: Line protocol on Interface POS1/1, changed state to down
06:20:51.819	SONET-4-ALARM: POS1/1: LRDI
06:21:02.131	SONET-4-ALARM: POS1/1: SLOS cleared
06:21:02.231	LINK-3-UPDOWN: Interface POS1/1, changed state to up
06:21:03.231	LINEPROTO-5-UPDOWN: Line protocol on Interface POS1/1, changed state to up
06:21:24.559	LINEPROTO-5-UPDOWN: Line protocol on Interface POS1/1, changed state to down
06:23:12.992	SONET-4-ALARM: POS1/1: LRDI cleared
06:23:24.680	LINEPROTO-5-UPDOWN: Line protocol on Interface POS1/1, changed state to up
06:23:25.704	CLNS-5-ADJCHANGE: ISIS: Adjacency to trd-gw (POS1/1) Up, new adjacency

Table 5.2: Router log from Tromsø-gw on the 7Th of May 2007

The Tromsø-gw's interface is up more than 2 minutes before the other end, and sets up the line protocol. Because Trd-gw does not have the interface up, it does not respond to the hello PDU, and Tromsø-gw has to change the state of the line protocol again to down.

When Trd-gw is connected and the interface is up, the line protocol follows. Then it takes almost 5 seconds before Tromsø-gw changes the state of the line protocol to up. And about 1 second thereafter the adjacency is established.

5.3.3 Flooded LSPs

As described in Chapter 5.1.2 are all LSPs flooded in UNINETT's backbone collected and presented by a served connected to Trd-gw. The tables 5.3 and 5.4 show the LSPs flooded due to the disconnection and connection of the Trondheim-Tromsø link.

Time	Life time	TLV type	TLV value
01:10:35.289777	1199	IS_NBR	vflag=0 -metric=12 1580.3900.0015.00
		IP_INT_REACH	- metric=12 network=128.39.47.96 mask=255.255.255.252
		IP6_REACH	- metric=12 up=0 external=0 sub_tlv=0 prefix=2001:700:0:6::/64
06:23:17.397068	1199	IP_INT_REACH	+ metric=12 network=128.39.47.96 mask=255.255.255.252
		IP6_REACH	+ metric=12 up=0 external=0 sub_tlv=0 prefix=2001:700:0:6::/64
06:23:27.401142	1199	IS_NBR	vflag=0 +metric=12 1580.3900.0015.00

Table 5.3: LSP collection from Trd-gw on the 7Th of May 2007

Time	Life time	TLV type	TLV value
01:10:35.491033	1191	IS_NBR	vflag=0 -metric=12 1280.3900.0082.00
		IP_INT_REACH	- metric=12 network=128.39.47.96 mask=255.255.255.252
		IP6_REACH	- metric=12 up=0 external=0 sub_tlv=0 prefix=2001:700:0:6::/64
06:21:02.567774	1191	IP_INT_REACH	+ metric=12 network=128.39.47.96 mask=255.255.255.252
		IP6_REACH	+ metric=12 up=0 external=0 sub_tlv=0 prefix=2001:700:0:6::/64
06:21:24.868176	1191	IP_INT_REACH	- metric=12 network=128.39.47.96 mask=255.255.255.252
		IP6_REACH	- metric=12 up=0 external=0 sub_tlv=0 prefix=2001:700:0:6::/64
06:23:24.993031	1191	IP_INT_REACH	+ metric=12 network=128.39.47.96 mask=255.255.255.252
		IP6_REACH	+ metric=12 up=0 external=0 sub_tlv=0 prefix=2001:700:0:6::/64
06:23:29.809280	1197	IS_NBR	vflag=0 +metric=12 1280.3900.0082.00

Table 5.4: LSP collection from Tromsø-gw on the 7Th of May 2007.

The LSP number and sequence number is of little interest in this study, and is not mentioned in the tables. It is important to understand that the time stamp in this LSP collection is the time when the server received the LSP, not when the originating router sends it! Because of that is the LSP from Tromsø-gw delayed compared to the LSP from Trd-gw.

It may seem smart to use the life time stamp to calculate the time of origin, but this field is decremented individually in each router on the path, making it impossible to use this information. The life time field in the LSP from the Trd-gw is decremented 1 s, while the life time field from the Tromsø-gw is decremented 9 s during the link failure, and 3 s after the link is brought back up. This implies that the life time field is decremented about 1 s at each router on the path.

	Packet loss [s]					
To→ **From↓**	*Alta*	*Narvik*	*Oslo*	*Svalbard*	*Tromsø*	*Trondheim*
Alta	—	N.A	8.3	N.A	N.A	N.A
Narvik	N.A	—	N.A	0	0	9.0
Oslo	6.1	N.A	—	N.A	57	N.A
Svalbard	N.A	0	N.A	—	0	9.0
Tromsø	N.A	0	8.5	0	—	9.0
Trondheim	N.A	6.5	0	6.5	57	—

Table 5.5: Periods of packet loss on the 7Th of May 2007, starting 01:10:34 am

5.3.4 Loss Time from Measuring Probes

Table 5.5 shows the observed periods of packet loss starting at 01:10:34 am when the Trondheim-Tromsø link is disconnected, and Table 5.6 shows the observed periods when the link comes back up again. Each probe only sends traffic to a subset of the other probes, and that is the reason for the not available (N.A) values in the table. The traffic to and from the measurement probe in Ny-Ålesund did not go via the affected link. The results from that probe are of no interest in this study, and is not presented in the tables.

By studying Table 5.5 it is observed shorter periods of packet loss in the direction south-to-north than opposite, except the flows from Oslo and Trondheim to Tromsø (57 s). The huge amount of packet loss is observed to these flows at similar occasions back to October 2006. A theory to these long periods with packet loss was a misconfiguration to the load balancing between Trd-gw and Harstad-gw3. A closer study of the event showed that the two flows are sent on the two different paths to Harstad-gw3. Because this seems like a rare event, these two observations of long periods with packet loss are not considered further in this report.

The shorter period of packet loss in the direction from Oslo and Trondheim to the other probes, implies that Trd-gw reacts faster to the failure than Tromsø-gw. Another reason may be the constructions of micro loops on the path between Narvik-gw2 and Tromsø-gw.

It is also observed packet loss when the Trondheim-Tromsø link comes up. Table 5.6 shows there was no packet loss in the direction south-to-north, except the flow from Oslo to Alta. This flow has a five times faster granularity than the other flows, making it more likely to lose packets. In this flow there was only one lost packet, which may be due to a small micro loop on the path.

The flows to Oslo and Trondheim experienced some packet loss, but have a slightly difference. The flows to Trondheim lost just one packet which most likely is because of a small micro loop between Narvik-gw2 and Tromsø-gw, or a short flapp to the Trondheim-Tromsø link after the adjacency is established.

Oslo experiences packet loss from Alta and Tromsø, but non from Trondheim. Because of the difference in the granularity and the interval of the sending of the

	Packet loss [s]					
To→ From↓	*Alta*	*Narvik*	*Oslo*	*Svalbard*	*Tromsø*	*Trondheim*
Alta	—	N.A	2.8	N.A	N.A	N.A
Narvik	N.A	—	N.A	0	0	1.0
Oslo	0.2	N.A	—	N.A	0	N.A
Svalbard	N.A	0	N.A	—	0	1.0
Tromsø	N.A	0	3.5	0	—	1.0
Trondheim	N.A	0	0	0	0	—

Table 5.6: Periods of packet loss on the 7Th of May 2007, starting 06:23:30 am

measuring PDUs, the amount of downtime is most likely the same for both flows to Oslo. The difference in the period of packet loss for the flows to Oslo and Trondheim, may be due to the load balancing between Tromsø and Trondheim, but it seems like the Tromsø-gw is a bit slow to update its routing tables.

5.3.5 Link Load and Packet discards

There was observed packet loss in small intervals during the whole period of maintenance work to the Trondheim-Tromsø link, but just one or two packets in a row. This is because of overload to the Narvik-Trondheim, Harstad-Bodø, Bodø-Mo, and Mo-Trondheim links as seen in Figure 5.1 on page 37. Figure 5.3 shows the load compared to the capacity on the Narvik-Teknobyen (Trondheim) link. The link has just a capacity of 155 Mbps, which is too low to manage the bursts.

Figure 5.4 shows there were packets discarded on the Narvik-Teknobyen link during the period of heavy traffic. These observations of small packet loss do not influence the convergence time, and is not interesting for the study in this report.

By comparing the two figures it is observed packet loss due to overload when the average load is about 80 % or more of the capacity on the link.

5.3.6 Comparison of Measurements in Case Study

As described in Chapter 5.2 is it possible to find the different components of the convergence time by comparing the results from the measurements.

5.3.6.1 Link Down - Bad news

This section describe the scenario when the Trondheim-Tromsø link was disconnected.

Failure Detection Component According to the measurement probes the failure starts at 01:10:34. This observation has an error margin ±500 ms due to

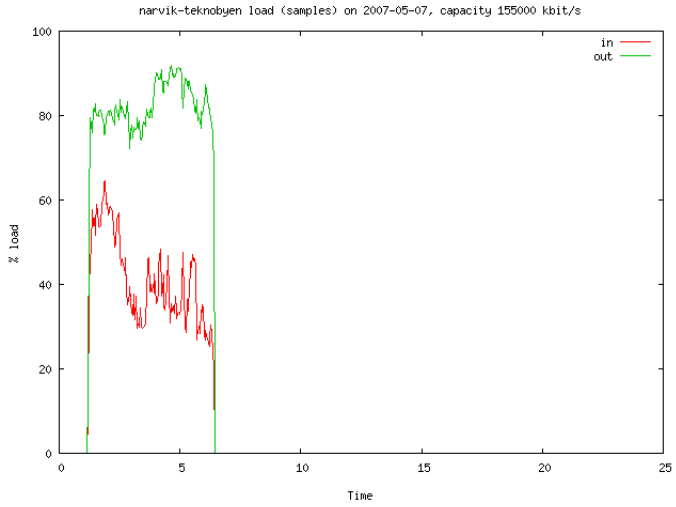

Figure 5.3: Load in the percent of the capacity on the Narvik-Teknobyen link on the 7Th of May 2007[31].

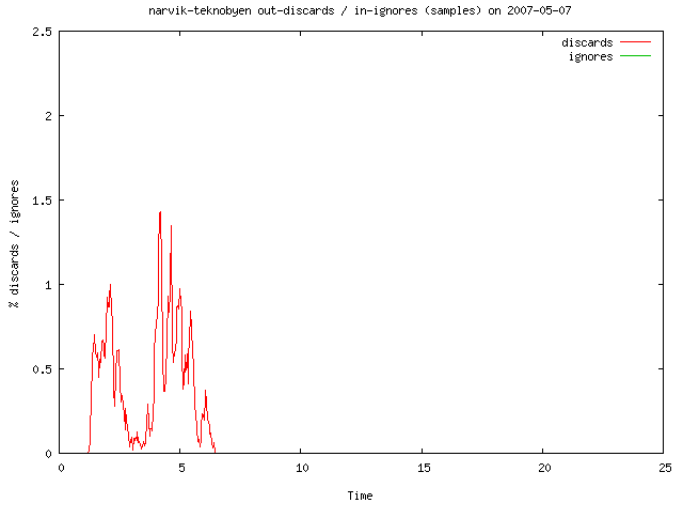

Figure 5.4: Packet discards/ignores on the Narvik-Teknobyen link on the 7Th of May 2007[31].

the interval of the PDUs. The router log from Trd-gw (Trondheim) changes the interface state (LINK-3-UPDOWN) to down at 01:10:34.983, while Tromsø-gw does the same 11 ms later. If the time stamp on the Trd-gw's LSP is compared to the router log (See tables 5.1 on page 42 and 5.3 on page 44), it is seen that the LSP is collected before the router change the state of the line protocol (LINEPROTO-5-UPDOWN). This implies that it is the "LINK-3-UPDOWN" which triggers the IS-IS to generate the LSP.

The failure is detected by the data-link layer between 0.5-1.5 s after the physical layer lost its connection. Because of the mentioned error margin from the measurement probes, it is impossible to state a more correct delay for the failure detection.

LSP Origination The router log from the 7Th of May does not include the LSP-generation history, which makes it even harder to state the exact delay from this component. The default value of 50 ms for the initial wait is configured, and makes it the smallest possible delay.

If the time stamp from the LSP flooded from Trd-gw is compared to the Trd-gw's router log, it is seen that the LSP is collected 306 ms after the failure is detected (See tables 5.1 on page 42 and 5.3 on page 44). The server collecting the LSPs and presenting the history is connected to Trd-gw, which makes the propagation- and serialisation delay negligible. The error margin lies in the bufferisation and IS-IS processing time.

Because of the mentioned lack of exact measurements, the exact LSP origination delay can not be stated more precisely than less than about 300 ms. Most likely is the delay close to 50 ms.

Flooding Because the LSP-generation history is not available, the accuracy of this analysis is poor. The same observations as for the LSP origination can be used to suggest the delay from this component.

The LSP from Trd-gw is received 306 ms after the failure is detected. If the LSP origination delay is about 50 ms, the flooding delay is 250 ms from the closest neighbor.

From the other end of the failed link it takes 497 ms before the LSP is collected (See tables 5.2 on page 43 and 5.4 on page 45). This gives a flooding delay just under 450 ms. This LSP has traveled more than 1200 km and passed trough at least five routers on the path from Tromsø to Trondheim.

Those two calculation gives the flooding delay from the shortest and longest paths in UNINETT's backbone. The delay may vary as the bufferisation and IS-IS processing times are far from constant.

SPT Computation As for the LSP-generation, the SPF history is neither included in the router log from the 7Th of May, which makes it impossible to state the exact delay this component introduce to the convergence time. However, the major contributor are the SPF and PRC delay parameters as mentioned in Chapter 5.2.4. Configured with the default values the SPF process

is delayed 5.5 s, and the PRC process 2 s, while the calculation itself takes less than 50 ms.

In this case were the Trondheim-Tromsø link failed, a full SPF calculation was required. The 5.5 s delay before the SPF process was run when the link failed, imposed packet loss for the whole waiting time. This makes the SPT computation a major contributor to the overall convergence time.

Updating RIB and FIB, and Distribution to Line Cards Due to the lack of the SPF history in the router log, these components is not possible to analyse. If the suggested values in Chapter 5.2.5 is used, the delay from these components are less than 400 ms.

5.3.6.2 Link up - Good News

This section describe the scenario when the Trondheim-Tromsø link was brought back up. The traffic was in the paths via Narvik and Bodø, but to limit the load to these links it was of interest to reroute the traffic back to the 2.5 Gbps link between Tromsø and Trondheim.

Repair Detection The available measurements make it impossible to observe the delay between when the physical layer is repaired, and the data-link layer detects the repair. The earliest observation in this up event is the data-link layer detection in the router log.

LSP Origination and Flooding First the Tromsø-gw changed the state on the interface to up at 06:21:02.231, but since the Trd-gw in the other end did not respond, the adjacency was not established at this first attempt. By comparing the router log and the LSP collection from this early flapp, the delay between when the data-link layer detects the repaired link and the LSP is flooded to the server in Trondheim is found. This delay includes the LSP origination delay. It is observed 336 ms delay for the up event, and 309 ms delay for the down event (See tables 5.2 on page 43 and 5.4 on page 45)

A few minutes after the incident with flapping link as seen from Tromsø-gw, the Trd-gw changed the state of the interface to up at 06:23:17.277. It takes just 120 ms from the data-link layer is triggered, to the LSP is collected (See tables 5.1 on page 42 and 5.3 on page 44). Again, this delay includes both the LSP origination and the flooding, which means the flooding is down to 70 ms!

The Tromsø-gw changes the state of the line protocol again to up at 06:23:24.680, and the LSP is collected 313 ms later (See tables 5.2 on page 43 and 5.4 on page 45).

At 06:23:25.689 Trd-gw establishes an adjacency with Tromsø-gw, but this time it takes 1.712 s before the LSP is collected. Tromsø-gw establishes the adjacency 15 ms after Trd-gw. Due to the second-wait parameter for the LSP-generation timer, Tromsø-gw waits 5 s since the last generated LSP before this next LSP is generated. This issue introduce extra delay for the flooding process.

	Time [s]	
Component	**Min.**	**Max.**
Failure detection	0.5	1.5
LSP origination	0.05	0.3
Flooding	0.07	1.7
SPT computation	5.5	5.55
RIB&FIB update/Distribution delay	0.01	0.4
TOTAL convergence time	**6.13**	**9.45**

Table 5.7: Summation of the convergence components from the case study

SPT Computation As for the "bad news", the SPF process just takes about 50 ms, but it is delayed for 5.5 s. The measuring probes detected a few packets lost at 06:23:30, about 5.5 s after the line protocols in both routers was up and running. Because of the delay in the synchronisation process of the LSDBs, small micro loops formed and caused packet loss.

Updating RIB/FIB, and Distribution to Line Cards As for the "bad news", this component is not possible to observe from the measurements. The suggested value of less than 400 ms is also used here.

5.3.6.3 Summing up the Convergence Components

The total convergence time is found by just adding together the different components described in the preceding sections. Table 5.7 shows the sum of the convergence time with both the lowest and highest observed or suggested values for the different components.

The lowest observed packet loss from the measuring probes is 6.1 s on the flow from Oslo to Alta, and the highest is 9.0 s on the flows to Trondheim (See Table 5.5 on page 46, not considering the 57 s losses to Tromsø). Compared to the calculated values in Table 5.7 are the observed values in conformity.

When the Trondheim-Tromsø link was brought back up there was observed 3.5 s loss on the flow from Tromsø to Oslo, 2.8 s loss on the flow from Alta to Oslo, and 1 s on the flows to Trondheim (See Table 5.6 on page 47). All these incidents are due to micro loops formed during the synchronisation phase of the LSDBs, and/or if the repaired link flapps after the new shortest path is calculated.

5.4 Consistency analysis

Before starting the test it is important to analyse the consistency when altering the parameters. A pitfall is the SPT computation versus the LSP flooding as mentioned in Chapter 3.5 on page 25.

5.4.1 Hello Parameters

The default hello parameters in the IS-IS protocol implies up to 30 s delay before an adjacency is torn down in case of a link failure. In stead there are detection mechanisms and timers in use in the lower layers, triggering the IS-IS protocol in case of a failure. Because of this the hello parameters in IS-IS are more or less unnecessary, and implies just overhead to the network. The overhead uses valuable CPU capacity, and may slow down the overall convergence time. Because of this it is of interest to keep the overhead as low as possible, and the hello parameters are preferred set to their maximum values.

The exceptions are odd node failures where the links seems OK, and switched networks. If a link between two routers is separated by two or more switches, a failure between the switches will not be detected by the routers. Both these exceptions are solved by introducing BFD as described in Chapter 3.2.1 on page 22. Because BFD is not yet available as a feature in commercial equipments, to days solution is to tune the hello parameters to speed up the detection process.

Table 5.8 gives a summary of the recommended values, and their pitfalls, for the hello parameters. The table shows both cases with low and high values. The low value is chosen if no other detection mechanisms are present, and the high values are chosen if e.g. BFD is implemented. If BFD is implemented and the high values are used for the hello parameters, the IS-IS hello PDUs is not really in use, and the consequence of the high value has no effect.

	Recommendations			
	Value		**Consequences**	
	Low	**High**	**Too low**	**Too high**
Hello Interval	{minimum} (-Implement BFD instead of tuning the hello parameters too low!)	65535 [s]	- May impose huge CPU load and increase the convergence time.	-Long delay in case of odd node failures and failures in switched networks.
Hello Multiplier	3	1000		
Hello Holdtime	$\approx$ 1[s]	$\approx$ 2[years]		

Table 5.8: Recommended values for Hello parameters.

5.4.2 LSDB update Parameters

The LSDB update process depends on many parameters controlling the SPT computation and the flooding of LSPs. All this parameters and their default values are described in details in Chapter 2.3.2.1.

Table 5.9 on the following page gives a summary of the recommended values for the LSDB update parameters, and their tuning consequences.

5.4.2.1 SPF and PRC computation

The SPT computation halts all other processes and is therefore a major contributor to the overall convergence time, if run to often. To keep the delay small in case of a single failure, and the CPU load small in case of flapping links, the exponential backoff algorithm is introduced. The goal is to compute the SPT fast in case of a topology change, but wait for consecutive LSPs before executing the SPT calculation in case of flapping links. A SPT computation is either a full SPF calculation or a PRC. In this setting they share the same constraints, and it is chosen the same values for both parameters.

Before the SPT calculation is run, all packets routed to the failed link or node are lost. Because of this it is of interest to run the SPF or PRC as soon as possible after the failure is detected. However, if it is a transient failure the SPT computation has to be run again, and the convergence process is further delayed. The initial delay is set to a very low value. Caution should be exercised when tuning the second- and max delay parameters. If a router receives a LSP with TLV changes during a SPT computation, the router will perform a new SPT calculation right after the first is finished. If new LSPs keep coming, the SPT computation may delay the convergence time more than necessary. The second wait parameter is set to a few ms higher than the initial delay, while the max wait parameter is set to more than 1 s higher than the initial wait to cope with transient failures.

To speed up the calculation process, the iSPF command described in Chapter 3.5 on page 25 is enabled. If the delay before the iSPF command is activated is set to low, the result of the SPF calculation could be wrong. A router with a wrong SPT would possibly impose routing loops and packet loss.

5.4.2.2 LSP generation and transmission

Prior to the SPT computation it is important that the LSP triggering the computation is sent to the router's neighbors. This could be a problem when small values for the SPF initial delay is used, because the SPF process preempt all other processes in the router. If the fast flooding command described in Chapter 2.3.2.2 is enabled, the LSP is flooded before the SPF calculation. This solves the problem in case of a permanent failure, but not necessary in case of a transient failure or a flapping link.

If a failure is repaired before the SPF calculation starts it is important that this information is flooded to the other routers before they start the SPF calculation. Otherwise it will introduce inconsistency in the SPTs, which would impose

	Exponential backoff			**Single parameter**	**Consequences**	
	max [s]	**int. [ms]**	**sec. [ms]**		**Too low**	**Too high**
SPF/PRC	1	10	100	—	-Delay convergence in case of flapping links.	-Imposes unnecessary packet loss during wait period.
LSP-generation interval	5	1	10	—	-Overload CPU in case of flapping links or multiple failures.	-Delay flooding, or no gen. at all in case of flapping links.
	(-Shorter than SPF/PRC to prevent routing loops.)					
LSP-MaxAge	—	—	—	65535 [s] (-Maximum value to minimize overhead.)	-LSPs purged from LSDB before refreshed.	-Delay in LSDB update if LSP is not refreshed.
LSP-refresh interval	—	—	—	65000 [s] (-Need to be shorter than MaxAge.)	-Overhead may delay convergence.	-LSPs purged from LSDB before refreshed.
ZeroAge-Lifetime	—	—	—	60 [s] (default)	-May purge LSPs too early.	-Delay before updating LSDB.
LSP-transmission interval	—	—	—	1 [ms]	-May overload the CPU and links.	-Delays the flooding.
iSPF	—	—	—	Enabled.120 [s] (default)	-May impose inconsistency in LSDBs and routing loops.	-The algorithmic optimization is never used which delays the SPF.
Fast flooding	—	—	—	Enabled.5 (default)	-Delay LSP flooding.	-Delay SPF calculation.

Table 5.9: Recommended values for LSDB update parameters

routing loops and periods of packet loss. It may seem tempting to set the LSP generation interval to a minimum value, but this would impose a tremendous overhead to the CPUs in case of flapping links. On the other hand; if the delay is set to high, the LSP will never be generated in case of a flapping link and there will be periods of packet loss until the link is stable. The rule of thumb is to set the LSP generation parameters to about 10 times less than the SPT parameters. The initial wait is set to the minimum value of 1 ms. The second wait is set a few ms higher, and the max wait about 1 s higher than the initial wait.

The default value of the LSP transmission interval is 33 ms. This value was decided in the early '90s to limit the load to the CPUs and links. Compared to the hardware performance today this value is unnecessary high, and impose a delay to the convergence time. Because of this is the LSP transmission interval set to the minimum value of 1 ms.

5.4.2.3 LSP Lifetime

The LSP-refresh interval decides the periodic LSP generation when the network is stable. The use of this parameter is to discard stale information from the routers databases if a node or link is withdrawn from the network. Ideally this information is unnecessary and imposes just overhead to the network, because all information regarding LS adjacencies should be flooded through the network.

If a LSP is not refreshed the MaxAge timer eventually reaches zero, and the LSP is discarded from the LSDB after a grace period known as the ZeroAgeLifetime.

The MaxAge parameter is set to the maximum value of 65535 s (ca. 18 hours). The LSP-refresh interval need to be less than the MaxAge, or else the LSP may be discarded from the LSDB before it is refreshed.

5.4.3 LSDB synchronization Parameters

The LSDB synchronization process depends on two parameters; the LSP-retransmit interval on P2P links, and the CSNP-interval on broadcast links. Table 5.10 shows a summary of the recommended values and their tuning consequences.

	Recommended value [s]	Consequences	
		Too low	**Too high**
LSP-retransmit interval *(P2P links only)*	1	-Ack. not received before retransm.	-Conv. further delayed.
CSNP- interval *(DIS only)*	1 (-Enable P2P links were possible instead of tuning parameter.)	-Overhead may overload CPU and links	-Synch. is delayed, and may impose routing loops.

Table 5.10: Recommended values for LSDB synchronization parameters.

5.4.3.1 LSP-retransmit Parameter

The LSP retransmit parameter is part of the reliable flooding mechanism on P2P links. When a router sends a LSP to its neighbors it waits a certain time for an acknowledgement from each neighbor. This delay is controlled by the retransmit parameter. If the sender does not receive an acknowledgement within expiration of this timer, the LSP is retransmitted. Ideally the LSP is acknowledged without any retransmission, but transient link or node failures and packet drops caused by overload may disturb the LSP exchange.

If the retransmit parameter has a high value, the convergence will be further delayed in case the LSP is lost in flight. However if it set too low, the LSP may be retransmitted before the neighbor has managed to send an acknowledgement. This will impose more overhead to the network, which may delay the convergence.

The retransmit parameter depends of the propagation delay between the neighbors, and the processing delay in the router. Queue in input and output buffers and SPT calculation is the major contributor to the processing delay.

In UNINETT the longest link is between Trondheim and Tromsø with a round trip delay about 15 ms. The backbone of UNINETT consists of 97 Level-2 routers with a main CPU of 200 MHz. A complete SPF calculation of such a network should take approximately 50 ms according to[13]. In a worst case scenario, queuing and consecutive SPF calculations may delay the acknowledgement further. Because of this the LSP-retransmit parameter is set to about 1 s.

5.4.3.2 CSNP-interval Parameter

Broadcast links do not have a reliable flooding mechanism like P2P links. Instead the synchronization of the LSDBs is controlled by the DIS which periodically broadcasts a CSNP. This is pure overhead to the network, but is a trade

off to achieve a simple synchronization mechanism.

A low value will make the convergence time fast in case of a link or node failure, or if a LSP is lost in flight. But a too low value may overload the CPUs and links, which may delay the convergence further. A high value decreases the overhead, but will delay the convergence. To speed up the convergence time, but keeping the overhead at an conservative level, the CSNP-interval parameter is set to about 1 s.

A better solution is to change the broadcast links into P2P links were possible. In UNINETT many P2P connections use Ethernet as the data-link layer. Ethernet is a broadcast media, but a feature in Cisco routers makes it possible to enable P2P links to the interfaces. If a P2P link is established to a Ethernet, the synchronization process is reliable as described in 5.4.3.1, and the CSNP-interval is not in use.

5.5 Test Lab

After studying different approaches to improve the rerouting in UNINETT, this chapter proves the results in a test lab. The test lab consists of four Cisco 4000 routers, running IS-IS, with 10 Mbps Ethernet and 100 Mbps Fast Ethernet interfaces. The four routers were connected as shown in Figure 5.5.

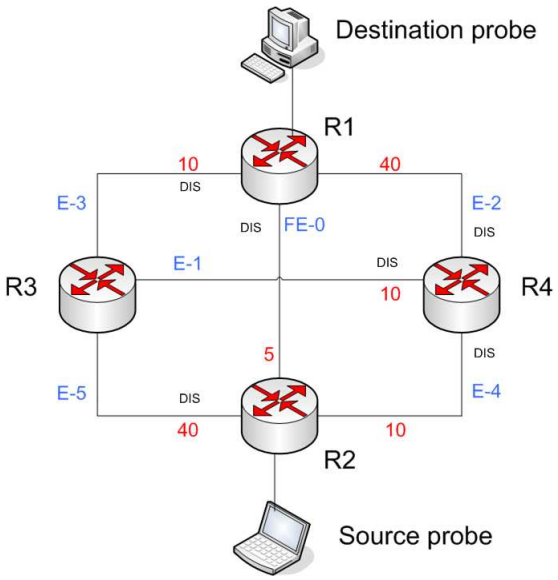

Figure 5.5: Test lab

The topology in the test lab is meant to reflect the topology of UNINETT in Northern Norway. R1 and R2 represent Tromsø and Trondheim, while R3 and R4 represent Harstad and Narvik. The measurements probes are connected to

R1 and R2 in an unidirectional configuration. The sender is connected to R2, and the receiver to R1.

The cost metrics decide the topology of a network. A low cost value is preferred compared to a high value when the routers calculate the paths . In this way the "logical" topology may differ from the "physical" topology. By setting the cost metrics on the E-2 and E-5 links to 40, these links are only included in the preferred path in case both FE-0 and E-1 fail. The FE-0 link gets a cost metric of 5, and is the primary shortest path. The rest of the links, E-1, E-3, and E-4, keep the default metric value of 10.

The cost metrics are chosen to simulate the traffic pattern seen in UNINETT:

No failures: The shortest path between the two probes is directly from R2 to R1 via the FE-0 link. Cost=5

FE-0 fails: The new shortest path is: R2-R4-R3-R1. Cost=30

E-4 fails: The new shortest path is: R2-R3-R1. (FE-0 still down.) Cost=50

E-1 fails: Introduces load balancing between R2-R4-R1 and R2-R3-R1.(FE-0 down, but E-4 up.) Cost=50

5.5.1 Default Values

At first the routers were configured with the default values to confirm the results with the current situation in UNINETT. The results from the first test is shown in Table 5.11.

In this test the Fast Ethernet link (FE-0) was disconnected and connected again 12 times. The third and sixth disconnection was performed shortly after the last connection (about a minute), and it is seen that the loss time is much shorter than the rest of the "down actions". This is because the router has an initial wait of only 2 s before a PRC, instead of 5.5 s before a SPF.

There are only two observations of loss when the link was brought back up, both caused by a flapping link as seen from router 1.

Action	Loss time [s]											
FE-0 down	8.2	8.7	3.5	8.7	8.1	3.8	8.6	7.9	8.5	7.8	7.6	7.3
FE-0 up	5.5	0	0.2	0	0	0	0	0	0	0	0	0

Table 5.11: Testing default values for FE-0

Observations of packet loss after repair:

1. (5.5 s) Router 2 floods a LSP about the up state of FE-0 before router 1 trigger a new down state. Router 4 runs a SPF before router 2 has established an adjacency with router 1, and there becomes a loop between router 2 and 4. About six seconds after the adjacency is established has router 2 updated its LSDB and the packet loss stops.

2. (0.2 s) Only one packet is lost. This happens about five seconds after the adjacency is established. Most likely a small loop between router 2 and 4.

If the two incidents with flapping links and the two observations of the PRC are not considered, the convergence time is about the same as observed in UNINETT; from about 7 s to almost 9 s. This is as expected, and in confirmation with the observed values from the case study in Table 5.5 on page 46.

The mean value of the 10 SPF observations is 8.1 s.

To confirm the rerouting in the test lab a second test with the default values was performed. In this test the different links were disconnected and connected in a sequence to check if the LSDBs were updated correctly, and to observe the loss time. It was observed that the shortest path was through router 4 and router 3 when FE-0 is disconnected, and the only path was through router 3 when E-4 was disconnected at the same time. E-4 was connected again and E-1 was disconnected. This imposed load balancing between router 3 and 4 as expected. Table 5.12 shows the different actions and the observed loss times.

Action	Loss time [s]
FE-0 down	8.2
E-4 down	7.2
E-4 up	0
E-1 down	6.7
E-1 up	0
FE-0 up	0

Table 5.12: Testing default values for the whole test lab

5.5.2 Tuned Values

After the tests with the default values was performed successfully, the parameters was changed to the recommended values from Chapter 5.4.

5.5.2.1 LSDB update Parameters

First only the parameters concerning the LSDB update was changed. It was run the same action sequence as for the default values, and the test was repeated

Action	Loss time [s]	
FE-0 down	2.7	2.3
E-4 down	1.3	1.8
E-4 up	0	0
E-1 down	1.2	1.3
E-1 up	0	0
FE-0 up	0	0.3

Table 5.13: Testing recommended values for only the LSDB update parameters

twice. Table 5.13 shows the observed loss times. It is seen from the results that the convergence time has decreased between five to six seconds compared to the test with the default values, and there is only one observed loop.

5.5.2.2 Hello Parameters

Before the next test the hello parameters were also changed to the recommended values, to see if it could speed up the failure detection. FE-0 was disconnected and connected again. Table 5.14 shows the observed results. The results show a further decrease in the convergence time, and it is obvious that the failure detection is a significant contributor to the convergence time.

Action	Loss time [s]						
FE-0 down	0.9	0.9	0.9	0.9	0.8	0.9	1.0
FE-0 up	0	0	0	0	0	0	0.2

Table 5.14: Testing recommended values for all parameters

To investigate the hello interval's influence to the failure detection, the test was run again with the parameter changed to 5 and then 2 seconds. The results are shown in Table 5.15. Compared to the results with the default hello values (FE-0 in Table 5.13) and the results with the recommended hello values (Table 5.14), the results from the tuned hello values show a decrease in the convergence time as the hello interval decreases. The minimal values for the hello interval gives a failure detection less than 1 s. The failure detection mechanism in the data-link layer (Fast Ethernet) needs more than 2 s before it triggers the higher layer (IS-IS).

	Loss time [s]					
Action	Hello interval 5 s			Hello interval 2 s		
FE-0 down	2.0	2.2	2.3	1.6	2.1	1.8
FE-0 up	0	0	0	0	0	0

Table 5.15: Testing tuned values for hello parameters

	Loss time [s]						
Action	SPF/PRC 1-100-1000					SPF/PRC 1-500-2000	
FE-0 down	0.9	1.0	1.0	1.0	1.0	1.3	1.3
FE-0 up	0	0	0.2	0.2	0.2	0	0.2

Table 5.16: Testing tuned values for SPT parameters

5.5.2.3 SPT Parameters

To study the SPT computation component, the SPF and PRC parameter were changed to 1-100-1000, and then 1-500-2000 (Max wait [s]-Initial wait [ms]-Second wait [ms]). The hello parameters were kept to the recommended values. The results are shown in Table 5.16.

Compared to the results from the test with recommended values (Table 5.14), it is seen a small increase in the convergence time as the parameters are increased. This is due to the increased delay before the calculation is performed.

A more interesting observation is the packet loss when the link is brought back up. 50 % of the transitions to an up state gives a small packet loss! Because of the increased delay before the SPF/PRC calculation is run, the LSDB update process is delayed which may construct micro loops. This is especially an issue if there are consecutive LSPs triggering a SPT computation. The exponential backoff algorithm will wait 1 s before the next calculation, and the synchronization of the LSDBs may be disturbed.

5.5.2.4 LSP-generation Parameters

To study the LSP origination components influence, the LSP-generation interval parameter was changed to 5-100-1000. The test was run with the SPF and PRC parameters set to 1-500-2000, and to the recommended values; 1-10-100. The results are shown in Table 5.17.

As expected increases the loss time compared to the test with recommended values (Table 5.14). The flooding of LSPs are delayed, which delay the convergence time.

In the second test the SPF/PRC parameters are set to a shorter value than the LSP-generation. The SPT computation is performed before the LSP is flooded and the content of the LSDBs becomes inconsistent. Because of the small size of the network used in this test, the contribution to the convergence time is small.

	Loss time [s]			
	LSP-gen. 5-100-1000			
Action	SPF/PRC 1-500-2000		SPF/PRC 1-10-100	
FE-0 down	1.2	1.3	1.0	1.0
FE-0 up	0	0	0	0

Table 5.17: Testing tuned values for LSP-generation interval

5.5.3 Summary of Results

The recommended values gives a significant improvement of the convergence time in the test lab. From 8.1 s with the default parameter values, to 0.9 s. with maximum tuning of the parameters. This gives a reduction of almost 1/10!

The most unexpected result from the test was the delay in the failure detection with the default values. The Fast Ethernet and Ethernet interfaces needed more than 2 s to detect a link failure. The tuning of the hello parameters imposes more overhead to the network, but it introduced no overload in the test lab.

Tuning the SPT parameters close to the recommended values gave small affect to the convergence time in case of "bad news", but contributed to a larger amount of packet loss in the case of "good news". Because of that it is not recommended to try to save CPU load and capacity by being "cheap" with the parameter tuning.

When tuning the LSP-generation parameters between the default and recommended values, there was only a small improvement in the convergence time close to the recommended values. This means the parameters could be set a bit lower than the recommendation, to reduce the overhead, without a significant deterioration in the convergence time. However, there was observed no problem with to much overhead in the test lab and the parameters were tuned to their maximum recommended value.

The tests show it is possible to achieve sub-second IGP convergence with a few changes to the IS-IS parameters!

5.5.3.1 Prevarication of the Test Results

The small scale of the test lab and the low traffic intensity on the links, made the load to the routers and links too low to observe any problems with overload. In a production network were the load to the links and nodes is higher, the overhead may be to high and cause overload. An overload to a node or link will cause packets to be discarded. This will introduce retransmission, and even more load to the network. In a worst case the overload will rise to a level were the network collapse, and all services are unavailable.

Because of the risk of creating a unstable network, the results from the test lab need more study before the recommendations are implemented in UNINETT.

5.6 Case Study VS Test Lab

This section discusses the observations from the case study compared to the results in the test lab. The questions are why, how, and what kind of information, the results from the test lab can give regarding the situation UNINETT.

5.6.1 The same Technology

The equipment used in the test lab is of the same technology as in UNINETT, except that the line cards in the routers used in the test lab is slower. However, this makes no difference in this setting because the serialisation and propagation delay is of less interest, than the IS-IS processing time. The main point in this study is the IS-IS routing protocol, which is the same in the test lab and in UNINETT's production network.

The observations from the case study and the results from the test lab with the default values is about the same. This means the basic knowledge and theory about IS-IS and fault handling discussed in Chapter 2 and 3 is correct, and can be used in the further study of improvement of rerouting.

5.6.2 The same Topology

First of all is it necessary to see the limit of the small scale network used in the test lab. There was no observed problems with overload in the test, but the load to the CPUs and the links was low. In a larger network were there is more competition for the resources, the overload may be a problem.

The topology of the test lab was chosen to reflect the situation of UNINETT in Northern Norway. By disconnecting and connecting the chosen links in the test lab, the scenario from the case study was reconstructed. The observed behavior from the test lab represented in this manner UNINETT.

5.6.3 The Components of the Convergence Time

By studying the behavior of the test lab it is possible to observe the different parameters influence to the convergence time. The results from the test lab give more information to further investigate the different components of the convergence time.

Conclusions derived from the test lab:

- The failure detection is a large contributor as indicated in the case study. Possible to improve by tuning hello parameters.
- The LSP origination delay is very small.
- The Flooding component is small, but does not give reliable information due to the small scale of the test lab.
- The initial delay of the SPF/PRC component is a major contributor as expected from the observations in the case study. Huge improvement when tuning parameters.
- The RIB/FIB update and the distribution delay are not observed due to the resolution of the available measurements.

The small scale of the test lab made the testing of the fast flooding and iSPF command worthless. The LSDB synchronization timers discussed in Section 5.4.3 were neither tested due to the small scale of the test lab.

The Ethernet connections used as P2P links can be enabled as P2P instead of the "default" broadcast medium. This will eliminate the extra pseudo node on the link, and the CSNP interval will be disabled. The overhead to the CPUs and the links will decrease due to the less processing load. The links in the test lab were enabled as P2P links, but again due to the small scale of the test lab there were observed no changes to the behaviour of the network.

Chapter 6

Conclusions and Further Work

This chapter sums up the conclusions from the thesis, and suggests research areas for further work. The results from the test lab implies it is possible to improve the rerouting in UNINETT, but further studies may find solutions to more and better improvement.

6.1 Conclusions

This section describes the results and conclusions of the thesis. The results and conclusion for each task foreseen in the problem description is presented separately.

6.1.1 Fault Handling and Micro Loops

Fault handling in IP networks with path restoration techniques has been studied. UNINETT is a network of that kind. Because there are no spare components to switch to in case of a failure, downtimes are unavoidable during the convergence phase. The goal is to make these downtimes as short as possible so the service unavailability as seen from the users is minimal. To achieve this it is necessary to implement a fast repair mechanism. A fast reroute mechanism like IPFRR[8] seems like a good solution. It protects against link or node failure by invoking locally determined repair paths. However, the advantage of fast repair is very small if not the micro loops during convergence are limited.

The phenomena with micro loops has been studied in Chapter 4. It has been concluded that it is a minor problem in UNINETT today. However, with the improvements possible by tuning the IS-IS parameters to the values recommended in this thesis, the micro loops may soon be a major contributor to the downtime in UNINETT. The requirements for service availability is getting higher due to the popularity of real time applications like VoIP and video streaming, and the

customers demand better QoS[2][3]. Because of this it is smart to look at the possibility to eliminate these micro loops. The oFIB method described in Chapter 4.3.7 on page 32 seems like a good solution. It is a good solution because of the compromise between a low delay of the convergence, and the cost of CPU and link capacity.

6.1.2 Downtime Contributors

The different components of the convergence time in UNINETT have been analysed in Chapter 5. It has been observed that the delay before the SPF calculation is run is the major contributor to the convergence time. The failure detection and the flooding process may also be large contributors. The difference between the minimum and maximum delay for those components is high, and implies that it is potential for improvement.

6.1.3 IS-IS Parameters affection to the Routing Scheme

IS-IS parameters have been studied in Chapter 2.3. It is seen that the default values imposes high delay in the convergence process. The IS-IS timers are the main reason that the convergence time in UNINETT is up to 10 s.

Ethernet connections used as P2P links can reduce the overhead to the CPUs and links by changing the configuration from the normal broadcast link to P2P link. This will reduce the load to the network and may speed up the routing scheme.

6.1.4 IS-IS Parametrization

Recommended values for the IS-IS parameters are described in Chapter 5.4. It is a huge potential in UNINETT to tune the parameters to improve the restoration performance. Especially the SPF timer, but also the hello timers and the flooding of LSPs. The flooding delay can be reduced if fast flooding is enabled.

6.1.5 The Topology of UNINETT

The topology of Northern Norway have been studied. It is not proposed any changes to the topology. The topology itself is thought to be a very small contributor to the rerouting times in UNINETT, but is subject for further study.

6.1.6 Failure Detection Techniques

The failure detection delay can be reduced with different solutions; tuning hello timers, tuning data-link layer timers, or implementing BFD. BFD is a better solution than tuning the hello timer too low, because it is less processor demanding than a high amount of hello packets. BFD is a research area as described in Chapter 3.2.1 on page 22, and is subject for further study. Tuning the data-link

layer timers may give problem with instability in case of flapping links, and is also a subject for further study.

6.1.7 Testing of Results

This thesis has documented that sub-second convergence is achievable in a test lab. The results from the test lab showed a reduction of almost 1/10 in the convergence time. This was achieved when the default values for the IS-IS parametrization was replaced with the recommended values from Chapter 5.4.

The methods from the test lab can be transferred to UNINETT, possibly without any compromise on the stability of the network. The prevarication of the test is due to the small scale of the test lab. There was no observation of problem with overload in the test lab with the recommended values, but this has to be studied in a larger scale before implemented in UNINETT.

6.2 Further Work

This section describes subjects of interest for further study to improve the rerouting in UNINETT further.

6.2.1 Tuning Data-Link Layer Timers

The case study and the results from the test lab showed that the failure detection is a large component of the convergence time. It is of interest to study the parameters in the data-link layer to see if it is possible to speed up this detection time, without the chance of instability in case of flapping links.

6.2.2 Implementation of BFD

Another possible solution to reduce the failure detection time, is to implement BFD between the end points of a link as discussed in Chapter 3.2.1 on page 22. Especially switched Ethernet connections are vulnerable to long delay in the failure detection, and would benefit of BFD. BFD does only use the CPUs in the line cards in the routers, and saves the main CPU of processing. This makes BFD a better solution than tuning the hello parameters too low, because the hello PDUs are processed by the main CPU in the router.

6.2.3 Implementation of oFIB and IPFRR

It is also of interest to study the possibility to implement fast repair and micro loop prevention mechanisms in UNINETT. If this is successful, sub-millisecond rerouting times may be possible! Work is in progress within IETF to specify a framework for IP fast reroute mechanisms[6]. The IPFRR mechanism [8] is interesting research material, and in addition to the oFIB mechanism described in Chapter 4.3.7 on page 32, it may be THE solution to rerouting in UNINETT.

Bibliography

[1] UNINETT. The Norwegian Research Network. June 2007. (url: `http://www.uninett.no/index.en.html`).

[2] S. Rai, B. Mukherjee, and O. Deshpande. IP Resilience within an Autonomous Aystem: Current Approaches, Challenges, and Future Directions. *Communications Magazine, IEEE*, 43(10):142–149, October 2005.

[3] S. Nelakuditi, S. Lee, Z.-L. Zhang, and C.-N. Chuah. Fast Local Rerouting for Handling Transient Link Failures. *IEEE/ACM Transaction on networking*, 15(2):359–372, April 2007.

[4] S. Bryant and M. Shand. Applicability of Loop-free Convergence. *IETF*, October 2006. draft edition.

[5] P. Pan, G. Swallow, and A. Atlas. Fast Reroute Extensions to RSVP-TE for LSP Tunnels. RFC 4090 (Proposed Standard), May 2005.

[6] S. Bryant and M. Shand. IP Fast Reroute Framework. *IETF*, October 2006. draft edition.

[7] M. Alicherry and R. Bhatia. Simple Pre-Provisioning Scheme to Enable Fast Restoration. *IEEE/ACM Transaction on networking*, 15(3):400–412, June 2007.

[8] A. Atlas and A. Zinin. Basic Specification for IP Fast-Reroute: Loop-free Alternates. *IETF*, March 2007. draft edition.

[9] Alex Zinin. Analysis and Minimization of Microloops in Link-state Routing Protocols. *IETF*, May 2005. draft edition.

[10] S. Bryant and M. Shand. A Framework for Loop-free Convergence. *IETF*, October 2006. draft edition.

[11] Pierre Francois, Clarence Filsfils, John Evans, and Olivier Bonaventure. Achieving sub-second IGP convergence in large IP networks. *SIGCOMM Comput. Commun. Rev.*, 35(3):35–44, 2005.

[12] NTNU. DAIM- Digital Arkivering og Innlevering av Masteroppgaver. June 2007. (url: `http://daim.idi.ntnu.no/`).

[13] Abe Martey. IS-IS Network Design Solutions. *Cisco Press*, 2002.

[14] Cisco Systems Inc. Cisco IOS IP Routing Protocols Configuration Guide, Release 12.4. June 2007. (url: `http://www.cisco.com/en/US/products/ps6350/`).

[15] ISO/IEC. Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model. *ISO/IEC 7498-1*, 1994.

[16] ISO/IEC. 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 (ISO 8473). *ISO/IEC 10589*, 2002.

[17] ISO/IEC. Information Technology – Protocol for providing the connectionless-mode network service: Protocol specification. *ISO/IEC 8473-1*, 1998.

[18] ISO/IEC. Information processing systems – Telecommunications and information exchange protocol for use in conjunction with the Protocol for providing the connectionless-mode network service (ISO 8473). *ISO/IEC 9542*, 1988.

[19] R.W. Callon. Use of OSI IS-IS for routing in TCP/IP and dual environments. RFC 1195 (Proposed Standard), December 1990. Updated by RFC 1349.

[20] C. Alaettinoglu, V. Jacobson, and H. Yu. Towards Milli-Seconds IGP Convergence. *IETF*, November 2000. draft edition.

[21] Cisco Systems Inc. SONET Triggers. February 2007. (url: `http://www.cisco.com/en/US/partner/tech/tk482/tk60`).

[22] Dave Katz and Dave Ward. Bidirectional Forwarding Detection. *IETF*, March 2007. draft edition.

[23] Dave Katz and Dave Ward. Generic Application of BFD. *IETF*, March 2007. draft edition.

[24] Dave Katz and Dave Ward. BFD for IPv4 and IPv6 (single hop). *IETF*, March 2007. draft edition.

[25] Dave Katz and Dave Ward. BFD for Multihop Paths. *IETF*, March 2007. draft edition.

[26] J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349.

[27] Pierre Francois, Olivier Bonaventure, Mike Shand, Steward Bryant, and Stefano Previdi. Loop-free convergence using oFIB, December 2006. draft edition.

[28] S. Bryant, M. Shand, and A. Atlas. Synchronisation of Loop Free Timer Values. *IETF*, October 2006. draft edition.

[29] UNINETT. Stager - Observations from measurement probes. (Under construction). June 2007. (url: `http://mi5.uninett.no/ arneos/routing/index.php?bookmark=1`).

[30] UNINETT. Historical data, LSP change statistics. June 2007. (url: `http://drift.uninett.no/cgi-bin/rtstat?net=uninett;area=backbone;proto=isis`).

[31] UNINETT. Graphical load map. June 2007. (url: `http://drift.uninett.no/stat-q/load-map/uninett, ,traffic,peak`).