UNIVERSITY OF OSLO
Department of Informatics

Monitoring Changes in the Stability of Networks Using Eigenvector Centrality

Master thesis

Ilir Bytyci
Oslo University College

May 22, 2006

UNIVERSITAS OSLOENSIS • MDCCCXI •

Preface

A paper composed of the findings on the thesis was written together with my supervisor Frode-Eika Sandnes, and is submitted to the ”7th International Conference on Parallel and Distributed Computing, Applications and Technologies” (PDCAT 2006) in Taipei, Taiwan, and is awaiting acceptance.

This thesis is the final work, carried out in order to fulfil the requirements of the Masters Degree on Network and System Administration, a programme organized by a cooperation between Oslo University College and the University of Oslo. During this programme I got hands-on experience with several network monitoring techniques and analysis, and got interested in ways to make them more efficient. After I was offered a list of theses topics to choose from, the topic related to monitoring stability of networks attracted my attention the most.

The target audience for this work are (but not restricted to) Internet Service Provider (ISP) operators that are involved in network monitoring and want to have an overall stability monitoring technique applied to their site. Network monitoring tool developers may use ideas behind this method to incorporate it with the existing conventional monitoring tools.

The main experimental analysis is based on a case-study, the network of UNINETT. The UNINETT Group supplies network and network services for Norwegian universities, university colleges and research institutions and handles other national Information and Communications Technology (ICT) tasks. UNINETT is owned by the Norwegian Ministry of Education and Research and consists of a parent company and four subsidiaries. The whole Group is located in Trondheim, Norway.

Acknowledgements

I would like to thank my supervisor Frode-Eika Sandnes for being so cooperative and bearing up with me during these last couple of months. It has been a great experience. Furthermore I would like to thank prof. Mark Burgess for his guidance throughout the Masters Programme at Oslo University College. A big thanks goes to Kyrre Begnum, a teacher, colleague, and a friend, for taking his time to have those discussions which several times made a difference for me. My appreciation also goes to the UNINETT contacts, Olav Kvittem and Trond Skjesol, who provided me with information and help when I needed it. In addition, I would like to thank my friends here at the College: Sven, Gard, Jon-Henrik, and Espen, my best friend Alban, and Arianit, the support and friendship of whom kept me going on.

Thank you,

Ilir Bytyci

Oslo, May 2006

Dedication

Dedicated to my loving family, who always supported me, and to my peace loving nation.

Abstract

Monitoring networks for anomalies is a typical duty of network operators. The conventional monitoring tools available today tend to almost ignore the topological characteristics of the whole network. This thesis takes a different approach from the conventional monitoring tools, by employing the principle of *Eigenvector Centrality*. Traditionally, this principle is used to analyse vulnerability and social aspects of networks. The proposed model reveals that topological characteristics of a network can be used to improve the conventional unreliability predictors, and to give a better indicator of its potential weaknesses. An *effective expected adjacency matrix*, k, is introduced in this work to be used with centrality calculations, and it reflects the factors which affect the reliability of a network, for e.g. link downtimes, link metrics, packet loss, etc. Using these calculations, all network backbone routers are assigned values which correspond to the importance of those routers in comparison to the rest of the network nodes. Furthermore, to observe how vulnerable each node could be, nodes are ranked according to the importance values, where the nodes with high ranking values are more vulnerable. This model is able to analyse temporal stability of the network, observing and comparing the rate of change in node ranking values and connectivity caused by the network link failures. The results show that the proposed model is dynamic, and changes according to the dynamics of the topology of the network, i.e. upgrading, link failures, etc.

Keywords: network monitoring, network stability, network reliability

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"Stability is unstable." - Hyman Minsky, an American theoretical economist

The Internet is "an electronic communications network that connects computer networks and organizational computer facilities around the world"[1]. Business activities are already dominated by various Internet-dependent applications. To be able to access services, end-users need to be connected with service provisioning locations, which are often physically far from them. Computer networks are complex systems composed of a group of hardware and software components which are connected together in order to help end-users achieve a certain goal [2]. These systems provide the Internet and intranet services to end-users to help them achieve their goals. Therefore, these systems need to be monitored, and repaired fast should any failures take place in the resource devices.

For simplicity, this work refers to computer networks as systems, and should not be confused with any other complex system[2].

1.1 Monitoring device failures

Systems interact with and are affected by the surrounding environment. Interaction between a system and the environment could be described in many ways: users accessing resources from the system, physical damage to links caused by nature disasters, loss of power supply, etc (see Figure 1.1). The interaction with the environment could affect systems to the extent that they fail to carry out the tasks end-users want them to. This is a problem for some applications which require uninterrupted services. Service instabilities in these applications would cause significant damages to users' needs and activities. Examples of applications that can tolerate very little, or almost no instabilities, are airport-controlling, business applications like stock markets, VoIP applications, real-time applications, etc [3].

[1] Merriam Webster's Dictionary, http://www.m-w.com, last time accessed in May 20, 2006

[2] For a full list of abbreviations refer to Section 8

Networks fail to provide uninterrupted services because of the nature of devices used in the system. From this information one gets the impression that there are no directly connected networks, but instead there are only ad-hoc networks [4]. System devices have a period when the are inactive. This is otherwise known as the devices' downtime periods. System downtime is the period of time system does not respond to service requests, because of one (or more) system element failures. Opposite to downtime period, a device uptime is the amount of time that the device is functional and serving the users' requests for its services [5].

Network topologies are composed of various devices. This thesis deals with central devices named routers, and throughout the thesis, the term node is used to refer to routers.

Device downtimes occur, be that caused during maintenance, or unintentionally by external factors. Among many, the causes are physical or software failures, typically caused by misconfiguration, maintenance, faulty interfaces, accidental fibre cuts, etc [6]. Devices used in computer systems have a period of time when they may not respond to service requests. Once device interfaces fail, links originating from those interfaces also fail. Additionally, the routing protocol configuration in the routers can affect system behaviour. Improper configuration, or bugs in the software that is configuring the device can cause the link or node downtimes.

In Section 2.4.1 when layered communication is discussed, in any of the layers described one could spot a source of instability. This shows the complexity of the task of identifying possible factors which affect network stability. To tackle stability issues one should have a picture of areas to focus on. Using a simple diagram one can identify only some of the possible sources of failure (see Figure 1.1). There have been attempts to explain causes of failure in an IP backbone [7], and this is discussed further in Chapter 3. The suggested fault tree is different and simple, in comparison to possible amount factors affecting stability of a network. Some of the failures expressed in the diagram can be easily prevented, as explained in Section 6.4. Other failures are difficult to monitor and prevent.

1.2 The issue of trust

Despite the failures occurring in the networks, services need to be guaranteed for the client. Service providers offer Service Level Agreements (SLA), as a way to guarantee customers a minimal provision level they are ready to offer. SLAs are contracts between a network service provider and a customer that specifies, usually in possibly measurable terms, what services the network service provider will furnish and what penalties will follow if the service provider cannot meet the established goals [8]. There is more discussion on SLAs in the section 2.2.2.

1.3 Motivation

Stability of service provisioning is one of the most important aspects of systems. Should there be no stability in the service provision, customers consider multihoming [9], i.e.

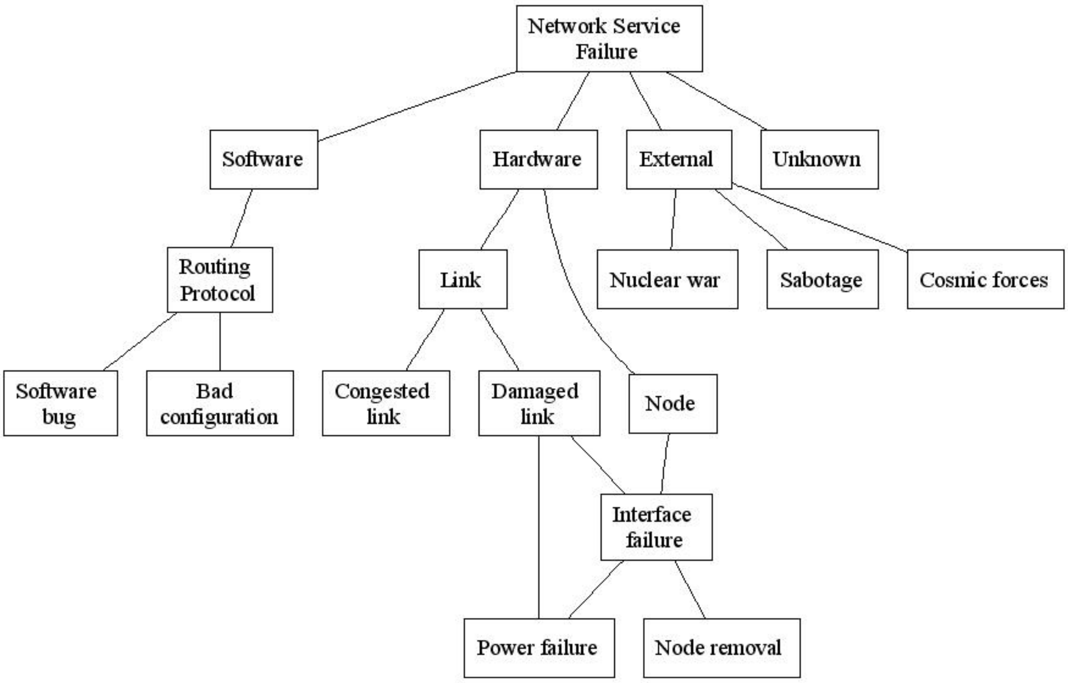


Figure 1.1: **Simple fault tree** - Analysis of some of the possible origins of failures, which affect the stability of the network

Client	*Provider*
telco	telco
ISP	telco
company	ISP
home user	ISP

Table 1.1: **SLA** - An example of actors bound to an SLA agreement. Telcos abbreviation stands for the telecommunication companies, which own the communication infrastructure. Internet Service Providers (ISP) then lease these infrastructures, to distribute their services to customers. Companies pay for services from ISPs.

considering redundant sources of accessing services. In the worst case scenario, the customers choose a more reliable source, and terminate the contract with the less stable source of services. Hence, the reliability and stability of service provisioning is very important to service providers.

Network service instabilities are unavoidable, and as such, failures are parts of the ordinary system activities which need to be dealt with. As a countermeasure, systems should prioritize fast-recovery of the network services once failures occur. Instabilities in network services are not always caused by link and node physical failures. The configurations of the routing protocol are also sources of instabilities. The network routing protocol is responsible for maintaining the network communication information between physically connected nodes. This one of the most important areas where stability can be improved. Network monitoring is carried out to identify what areas are suffering from instability.

1.3.1 The need for more effective monitoring technique

Monitoring the network for failures in its links and nodes, and identifying anomalies caused by them, is a typical duty of a network operator. Network monitoring tools are widespread, from many commercial to free tools. Using free and efficient network monitoring tools like *Nagios* [10], *Munin* [11], or *Rtanaly* [1] has been shown to be useful for monitoring the state of networks.

To the best of our knowledge, the conventional network monitoring tools tend to almost ignore some network topology characteristics and instead receive only the reliability information of individual nodes (routers) and links (lines of transmission). In networks where different types of nodes are connected through various types of links, including information on the network topology characteristics is important. The conventional network monitoring techniques do not consider how central i.e. how important is the observed node in comparison to the other nodes surrounding it. Importance of nodes has to do with the centrality of that node, and the weight it carries in distributing or forwarding traffic to the surrounding nodes. Clearly, a high-capacity node that is connected to many nodes through high-capacity links is more important than a low-capacity node connected to a only few nodes through low-capacity links.

The above mentioned details need to be a part of the stability monitoring model, so that recovery is prioritized, according to the level of instability and importance of nodes. By considering such network characteristics, a more effective view on the status of the network is obtained. To be able to improve the stability there is a need for a model that identifies the areas that suffer from instability. Analyzing the rate of change in ranking of nodes gives a picture of what nodes could be causing instability. Furthermore, monitoring the source of errors and their duration gives insights to a better understanding of the status of the network stability [6,7]. There is need for uses state of the art methods for locating network stress points using eigenvector methods, and combines these with probabilistic estimators for network reliability.

In earlier work, it has been shown that the most highly connected and central nodes in a network are also vulnerable in a number a ways [12]:

- They channel the most traffic, assuming that all nodes are responsible for generating approximately equal amounts. They, therefore, experience the most stress.
- Eliminating them would make the biggest impact on the network structure and cause the most problems.
- They are obvious targets for attack by malicious parties.

1.3.2 The need for techniques to improve stability

The proposed model introduces a different network monitoring strategy, which considers more aspects when analyzing the gathered network logs. At the same time it serves as an analysis tool to find ways of improving stability of networks. Having a clear hierarchy and being able to monitor the dynamics of stability some of the advantages that this model offers. To improve the stability in networks several suggestions were given in Section 6.4. Some suggestions are simple, and most of the today's network operators are well aware of those measures, but nevertheless, they still are worth repeating. Among some of the ways stability could be improved are by carefully addressing these issues:

Power supply - should be secured and redundant,

Human access - policies on human access rights should be enforced,

Fault tree - identify possible sources of failures, based on historical data,

Data collection - collection of various network logs (e.g uptime of links and nodes, packet loss, delay information, routing protocol transactions on link states, etc) from clients spread around the network to be used for the monitoring tool

Monitoring - apply network monitoring tools that are capable of sensing instabilities (for e.g. a tool that uses the model proposed in this thesis),

Tuning - routing protocol default configuration could be tuned, by changing parametres in the routing protocol, to speed up the convergence of the system,

Traffic Engineering (TE) - apply redundant protection to routing protocol, using Multiprotocol Label Switching - MPLS [13], Netscope [14], etc

Routing protocol convergence refers to the time required by a network to react, once a failure of a link or a router occurs, and recover to a stable state [15]. Saying that a system has a fast convergence time means that even if the system falls in a instable state, it recovers fast to a stable state, without causing damage to the service provisioning level. Thus, faster convergence significantly affects the stability of network services. Due to several delays in failure detection and propagation and new route recomputation, it may take tens of seconds to minutes after a link failure to resume forwarding of packets in that link [3]. This is unacceptable for services that need persistent services.

1.4 Objective

In recent years, network analysis has led to a variety of methods for modelling probable behaviour of networks, including eigenvector methods [12, 16] and graph invariants [17]. In this thesis a new model for observing overall network stability is proposed using a similar approach. This model measures the connectivity of the network, and identifies the most important nodes in the network and ranks them by importance. In the first part of the analysis the value of connectivity of the network is analysed over time to see indications of change in that value. If the connectivity value decreases significantly that is a sign of instability. The analysis follows with the node rank values are analyzed to identify what nodes are affected. A live running network topology and its data logs are used as a case study. To make it useful for stability monitoring, i.e. looking at how stability of network changes, node ranking is calculated not only using a single parametre, the number of neighbours one node has, but several factors, like link uptime percentage, link metrics, packet loss, etc.

This work also explains the stability issues in the Intermediate System to Intermediate System (IS-IS) [18] routing protocol, a common interior gateway protocol used by ISPs and telecommunication companies, and is quite similar to the Open Shortest Path First (OSPF) routing protocol [19]. IS-IS protocol is used by the network service provider, UNINETT [3], the network topology and information of which are used as a case study for the analysis in this paper. IS-IS is a link-state protocol, which means that it routes packets in the network based on the information gathered from the state of the links. The default configuration of the IS-IS routing protocol can be tuned to speed up the network routing convergence, once failures take place in network links or nodes. Several network protocol tuning methods have been suggested [3,15,20,21] which are discussed in detail in Section 3. There is need for careful analysis on these methods. Using the findings in the mentioned methods, ways of improving the stability are suggested (see Section 6.4). A faster routing convergence means a higher availability of routes.

1.5 Thesis structure

This section of the thesis explains the way the rest of the document is structured:

Chapter 2 - This chapter explains the background on which the rest of the document relies on. It opens a discussion on the network service stability issues, SLAs, as well as routing terminology and concepts. Chapter 2 is a thorough and detailed chapter, so the readers unfamiliar with the terminology and concepts get the information needed to understand the rest of the thesis. Users who do not need this background information should skip this chapter and continue with Chapter 3.

Chapter 3 - This chapter analyzes the work that is related to the topic discussed in this thesis, i.e. issues related to network stability and availability.

Chapter 4 - This chapter holds the explanation of the theory behind the stability monitoring model that is suggested in the thesis.

Chapter 5 - The Methodology chapter includes analysis on the origins of network failures, and explains the proposed a model for monitoring stability changes and decribes the experiments to be carried out in Chapter 6.

Chapter 6 - Results and Analysis chapter explains the data analysed and hypothesis posed in the Methodology chapter. It also contains a subsection with tips on how the stability of networks can be improved. Last section includes discussion and tips on how the overall network stability can be improved.

[3] http://www.uninett.no/ last time accessed in May 20, 2006

Chapter 7 - Conclusions and discussion chapter gives a summary of the thesis.

Appendix - This chapter holds the supplementary tables which are a part of the analysis and findings throughout the thesis. Additionally, the scripts used during the analysis and data gathering are shown, each of them with the place it is referred to in the text.

Chapter 2

Background

2.1 Scientific methods

This chapter is detailed and is intended for the readers who are not very familiar with networking terminology, and system analysis. The readers not belonging to this group should continue reading in Chapter 3.

As Burgess [2] defines it, ”the principle aim of science is to uncover the most likely explanation of the observable phenomena.” Systems change, and there is a cause for every change. This is known as the *cause and effect* law. In science it is very important to limit the scope of what causes can be monitored and what not. It is also important to not oversimplify the experiment to the extent that it does not make sense to analyse. In science only idealized or simplified environment can be observed. This is called *system modelling*. A system model represents the simplification a scientific experiment needs in order to approximate the understanding of the causes behind effects. Models are crucial to develop in order to ”interpret empirical data and motivate experiments and data are needed to substantiate theories or to inspire models” [4]. Experimental observation is the first and the main element meaningful experimentation should be based on. Observed evidence should then be compared with theory, to be able to approximate a conclusion.

In order to understand systems, measurements should take place. During measurements collected data during regular intervals are represented as *time series* or histograms. Time series show measured values and time at which the measurements were made. Histograms count the numbers of measurements that belong to a certain group in the data values [2]. This way one can observe any patterns in the plotted data, and deduce causes of such behaviour. A meaningful scientific deduction about a phenomenon must be supported by scientific modelling and theory, which is also a part of this thesis.

In order to further understand the system behavior, it is of interest to understand the rate systems change. The way the observed system calculations change is shown in the Section 6. Systems are (temporarily) stable as a result of a compromise between its *freedoms* and *constraints* on its changes [2]. Freedoms to change refer to the potentials of a system to change. Constraints are there to limit the amount of change allowed in the systems. Typical constraints are physical limitations of devices, different rules, protocols, etc. Suggestions and tuning proposed in Chapter 6.4 are some of the constraints that could be added, to prevent the systems' freedoms to change indefinitely, beyond the wanted stable state, defined by a system policy.

2.2 Stability and availability of systems

Modern computer systems are becoming increasingly complex, due to the complicated functions the network devices provide, and diversity of vendors which produce those devices. As a result, quantifying the stability of systems in one variable is also a complex issue, if not impossible. Stability of a system said simply is the firmness or the ability of the system to withstand the forces from the environment surrounding it, which constantly tend to change the state system is in. The sources of instability are various, and not only originating from the machines, but from human errors as well. Humans are an integral part of the systems today, therefore often computer networks are referred to as human-computer systems. Thus, as long as humans interact with the system, long-term stability of systems is not feasible. Not even medium-term stability is possible. A satisfactory duration of the stable state of the system is one during which the needs of end-users can be satisfied.

A stable node communication is the core element of the overall network stability. Nodes should be able to communicate between each other, even if any failure occurs in the system. This ability of a network is called *fault tolerance*. Fault tolerance of systems, and together with that its availability, are increased by providing redundancy of links. Redundant links between nodes ensure that in case of failure of one link between two respective nodes, the communication still can take place in the other redundant link. Often the redundant links are hierarchically classified according to their capacities. Metrics are assigned on the links to indicate the link characteristics [21, 22]. This way the links with lower metrics have a higher capacity and a shorter delay value, and the other way round. The link with lower metrics is prioritized to forward traffic. To improve the efficiency and to prevent congestion, load balancing is applied, so that not only the link with lower metrics is loaded, but traffic is forwarded through other redundant links as well. There should be a balance between the cost of provisioning several redundant links, with the increase in service availability they provide. One should be aware of the change in complexity of the system after redundant links are provided.

2.2.1 Service availability

According to [23] the term available describes a system that provides a specific level of service as needed. Availability is generally understood as the period of time when services are available or as the time required for the system to respond to users. The methods used to quantify the availability in a network are the so-called *the percentage* method and *the defects per million* (DPM) method [24]. As their names indicate, the percentage method calculates the percentage of network service uptime, say 99.9%.

This method is mainly used to give yearly estimations of downtimes. The number of ″nines″ determines the number of minutes, hours, or days per year when services are down. Looking at a calculation made by [24] we see that for 99.99% availability rate there is 1 hour downtime per year, for 99.9% - 8.5 hrs, 99% - 3.5 days, 90% - 36.5 days. Obviously the number of ″nines″ is important. DPM on the other hand has the ability to track more reliability issues than the percentage method. DPM is used to ″measure partial or full network outages″, the hours during which the device were operating, etc [24].

To analyse better the methods mentioned above several calculations are needed. Such are Mean Time to Repair (MTTR), ″is the amount of time (on average) that elapses between a network failing and the network being restored to proper working order″ [24], and Mean Time Between Failures (MTBF). Using these two calculations, one derives the formula for availability:

$$Availability = \frac{MTBF}{MTBF + MTTR} \tag{2.1}$$

Diot et al [7] define service availability from a source to a destination in an IP network refers to the ability of the network to deliver IP packets from the source to the destination. Port availability is the uptime of a single network element i.e. the hardware by which the customer attaches to the ISP's network to retrieve services. Path availability refers to the existence of physical connectivity between the points.

Network topology, that determines the number of alternate paths between two points, and whether they are link/router-disjoint [7], is another very important factor which affects IP service availability. Single points of failure should be avoided, when possible. Physical path diversity in IP to physical layer mapping is a very important consideration for service availability [25].

2.2.2 SLAs and their metrics

Some ISPs do not provide any SLA to their customers, but this is changing, because the issue of trust is very important when two parties are involved in a service exchange. SLAs attempt to establish a trust relationship between the service provider and service ″consumer″. Should the terms not be met by the service provider, the ″consumer″ is provided with an SLA which includes a way to get compensation for the loss caused by the underprovisioned services. There have been attempts, such as [26], to quantify the service availability as an SLA metric.

SLAs by today's ISPs are based on three metrics: loss, delay and port availability [26]. While loss and delay can be measured and relatively guaranteed for, port availability is difficult to quantify through SLAs. A typical Internet Service Provider (ISP) may be able to guarantee a loss of less than 1% end-to-end delay of 55 msec (within continental USA), and port availability of 99.9%. Availability 99.9% may not be sufficient for a telephone network, which requires ″five-nines″′ i.e. 99.999% availability [26].

2.3 Computer Networking terminology

Terminology used in networking, as well as understanding the network communication in a layered fashion is important to understanding the topic discussed in the thesis. Misconfigurations and failures coming from any of the layers in the computer networking architecture and protocols are only some of the possible sources of instability in the network.

In Webster's Dictionary[1] a computer is defined as a "programmable electronic device that can store, retrieve, and process data". In the world of computing, "the old model of single node serving all of the organization's computational needs has been replaced by one in which a large number of separate but interconnected nodes do the job. These systems are called computer networks" [27]. Sharing the costs of a leased line of communication, utilizing the capacity available, easily sharing content between participants in the network, are only some of the reasons why computer networks developed with such a speed. Furthermore, networks represent collections of people or devices that share resources in an attempt to achieve a common goal.

Today's modern computer networks are systems that evolves in time with a rate of change [2], and interact with the environment surrounding them. Instabilities caused by human factor may be the most difficult one to monitor and prevent. The fact that interaction with environment takes place, and humans being the core of the environment, it means that instability is unavoidable.

In computer networks inter-connected nodes could be routers, switches, hubs, end-user computers or devices, etc. Graphs are used to represent the abstraction of network links and nodes. A graph is a pair (X, γ) that consist of a set of nodes X and a mapping $\Gamma : X \rightarrow X$, formed by the arcs or lines between the points $x \in X$ [2]. Nodes are connected through links, which enable the transmission of information from one node to another. A node which has only incoming flows to it is called a sink node. The node which has only outgoing flows is called a source node. A node which has both incoming and outgoing flows is called a relay node. Depending on the directions of the flow identify systems in which only a supports transmission in one direction per period of time, called half-duplex links, and systems that support a two-way communication at once, called full-duplex links. The latter is supported by most of the links nowadays, and is found in the network IP backbone links.

Unregulated systems have the freedoms to change indefinitely. Unrestricted change in system could be harmful to the stability. Instead, to prevent instability, regulated systems make use of several constraints. Network traffic cannot be regulated by itself, so rules of conduct, such as protocols, should exist. According to [2] a protocol is a standard of behavior, or a strict rule of conduct that ensures that one part of a system is able to cooperate with another, and the integrity of the process is maintained, that is, information is not lost or misunderstood.

[1] Merriam-Webster's Dictionary http://m-w.com

2.4 Network Communication

Before being able to exchange information, the parties involved should be either directly or indirectly connected. Not necessarily should the nodes be neighbouring nodes, but in order to communicate there should be intermediate nodes which are able to forward packets to desired destinations. This is the first step before any communication can take place. The real information communication is not this simple. Instead communication is carried out in several steps in a layered fashion, in the so-called vertical and horizontal communications. Failures could occur while this communication takes place, thus it is important to mention possible sources of failures coming from these layers.

2.4.1 Layering concepts

In computer networks information transmission is done in a layered fashion. Information across a network is transmitted in different-sized (depending on the protocol) packets. These are called Protocol Data Units (PDU). Depending on the protocol which is used the transmission passes through 7 layers of International Standards Organization (ISO) Open Systems Interconnection (OSI) model, or 5 (sometimes 4) layered Transmission Control Protocol/Internet Protocol (TCP/IP) model. The latter is the de facto standard for transmitting data over networks.

The International Organization for Standardization (ISO) was formed to develop standards for data networking. The Open System Interconnection (OSI) protocols represent an international standardization program that facilitates multi-vendor equipment interoperability [28]. As it can be seen from Figure 2.1 the Application Layer of TCP/IP carries the same services as the Application, Presentation and Session Layers of OSI Model. TCP/IP model does not have a session and presentation layer at all. Transportation layers are the same in both models. Network layer of OSI Model corresponds to the Internet layer of TCP/IP model. Data Link and Physical layer of OSI Model are represented by Network layer in TCP/IP model (See Figure 2.1).

Service Access Point (SAP) are points located in the borders between layers. These locations are where one layer provides a service for the layer below, or is provided a service by the layer above. These are possible sources of instability, should these points be corrupted, or inconsistent information is passed to them. Typical SAPs are port identifiers in User Datagram Protocols(UDP) and Transportation Control Protocol (TCP), which are explained more in Section 2.4.1. As the Integrated IS-IS protocol discussed in this thesis is created by OSI, but supports the TCP/IP protocol (the defacto Internet protocol) then the latter will be explained shortly.

Application Layer

The application layer contains all the higher-level protocols, the client and server programs, such as TELNET (virtual terminal), file transfer (FTP), electronic mail (SMTP), Domain Name Service (DNS), one of the most famous protocols used for retrieving pages on the World Wide Web (HTTP). This is typically the layer that the end-user

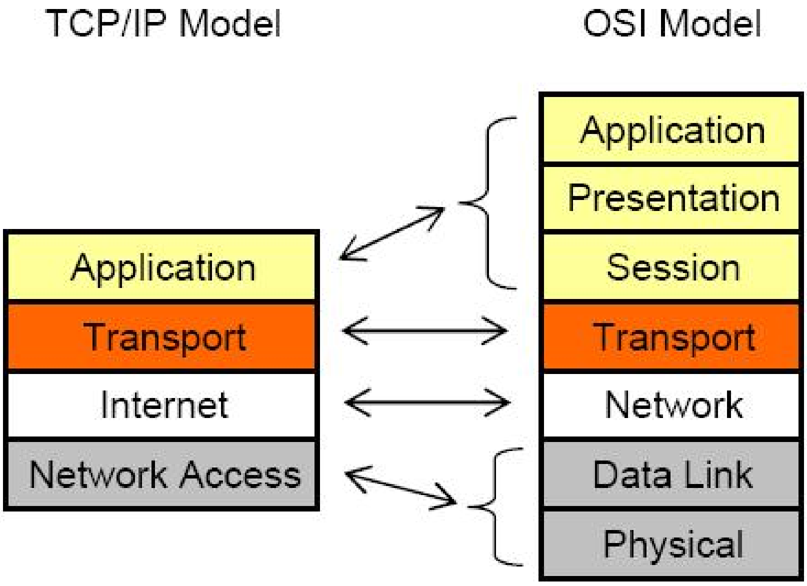


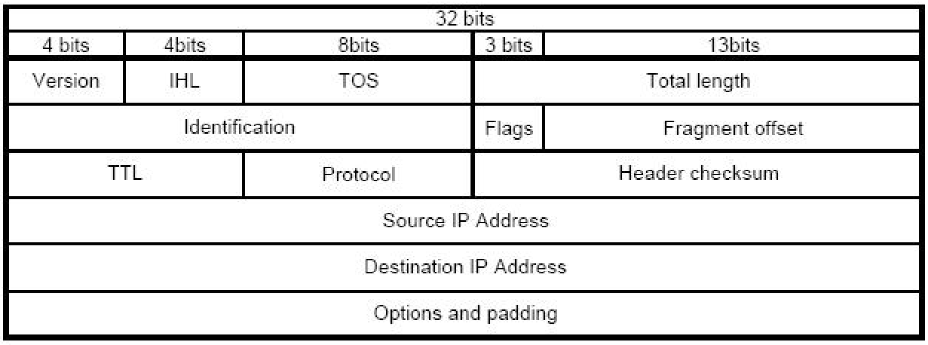
Figure 2.1: **ISO OSI vs TCP/IP Reference Model**: The representation of layers, and the comparison between two reference models

interacts with, and initializes requests for services. Once a message or a request for some service is created it is passed on to the next layer. Clearly, a faulty application could be harmful to the stability of a communication, and even further to the overall network. The typical applications that are known to cause such damage are computer viruses.

Transport Layer

The transport layer is designed to allow peer entities on the source and destination hosts to carry on a conversation. End-to-end protocols are defined here, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and other service ports. TCP is a reliable connection-oriented protocol that allows a byte stream originating on machine to be delivered without error on any other machine in the Internet. It fragments the incoming byte stream, originating from an application, into discrete messages, called segments, and passes each one onto the internet layer. During segmentation it appends(encapsulates) a TCP header to the message segment. Connection-oriented services first have to establish a connection between two end-nodes that want to transmit data.

On the other hand, UDP is unreliable, connectionless protocol for applications that do not want TCP's sequencing or flow control, and wish to provide their own. It is used for client-server type request-reply queries in which prompt delivery is more important than accurate delivery. A host utilizing a connectionless service does not need an established connection in order to send or receive data. As routing update packets should be small and when speed matters, UDP is better suited for transmitting routing information. This is more discussed in the next parts of this chapter, where



32 bits				
4 bits	4bits	8bits	3 bits	13bits
Version	IHL	TOS	Total length	
Identification			Flags	Fragment offset
TTL		Protocol	Header checksum	
Source IP Address				
Destination IP Address				
Options and padding				

Figure 2.2: **IP Packet fields** - Sizes and names of the fields in an IP packet. Some unused optional fields could be used for special labelling of packets, for better routing, as shown in 6.4.

routing and routing protocols are explained.

Internet Layer

Internet layer keeps all the network architecture together. Its role is to define how packet traffic is regulated, and permits hosts to insert packets into any network and have them travel to a specified destination [27].This layer is where network packet routing is done, and the most important layer, where stability can be improved, or where it can be compromised. Packets could arrive in different order, but since they are accompanied by a identification field, they are re-ordered at the receiving end-host. If a packet fails to arrive in the specified destination, it is retransmitted. The Internet layer defines an official packet format and protocol called Internet Protocol (IP). Every packet is addressable using IP addresses. The device which has the main role in ordering the traffic in this layer is called a router. A router [2] is a device that forwards data packets along networks. The process of packets being able to pass through intermediary nodes between sender and receiver is called routing. Routing is carried out based on routing tables, which are either statically or dynamically created tables containing information on the best available paths to the receiving node. This layer also provides flow control, segmenting and de-segmenting packets, and error control. Some IP packet fields are self-explanatory, but some are clarified below.

- Version indicates which IP version, 4 or 6 is used.
- Identification labels a certain fragmented packet, and whether it belongs to a datagram.
- Flags field explain whether the IP packet is followed by other fragmented packets, or whether it should not be fragmented.
- TTL (Time to Live) is the period of time before the packet is dropped.

[2] http://www.webopedia.com/TERM/r/router.html last time accessed in May 20, 2006

- Protocol specifies what upper layer protocol does the packet carry, for e.g.TCP, UDP, etc.

As will be shown later in the Section 6.4 the optional fields of IP could be used to use new technologies in finding best paths to transmit packets.

Network Access Layer

The data link and the physical layer of the OSI stack is incorporated in this layer. Data link layer provides a well-defined service to the Internet layer determining how bits of the physical layer are grouped together into blocks called frames. It also deals with transmission errors, and regulates the flow of frames so that slow receivers are not overwhelmed by fast senders. A frame is encapsulated using a frame header and tail, which is added to the packet, in order to make it distinguishable when inserted in a communications link. A frame is addressable by Media Access Control (MAC) address, which is an address that uniquely identifies each node's hardware.

The network access layer deals with the physical aspects as well. This is where the actual communication occurs, and it is where raw bits are transmitted over a communication channel. Physical damages of links or nodes are rare, but yet possible. Links and nodes could also not function for some period of time, as the DPM factor explained earlier. This is a source of instability which should not be ignored, and measures should be taken to apply further protection of links and nodes.

2.4.2 Vertical and horizontal communications

This communication is also referred to as logical and actual flow of information. Each layer, *n*, provides services to the layer above it, *n+1*, and receives services from a layer below it, *n-1* (if there is one). The main terms here are encapsulation and decapsulation of data, as shown in Figure 2.3. The actual communication (as shown with the dashed lines) passes through layers, and each layer encapsulates its header accordingly. Only then can the message be injected in the communication link to be sent to the receiving node. After the message is routed to the receiving node, it is decapsulated, passing through the layers as shown below. In the end it reaches the receiving application which is capable of reading the sent message. To applications and layers it seems as if the communication is direct, but that is only the logical communication [27].

Packets can be addressed to one or more receiver nodes. Thus, there are networks which are unicast, multicast, and broadcast. *Unicast* networks identify only one receiver of the information, and no other but that node in the network can decapsulate and read the content. If one piece of information is sent to a selected group of receivers, then that network is called *multicast*. When the information sent in the network is spread to all nodes in the network, then that network is a *broadcast(flooding)* network. Typically link-state routing protocols use flooding to send the routing information throughout the network. Thus it is important that the packets are small, and are distributed as fast as possible. For this matter, UDP packets are more suitable.

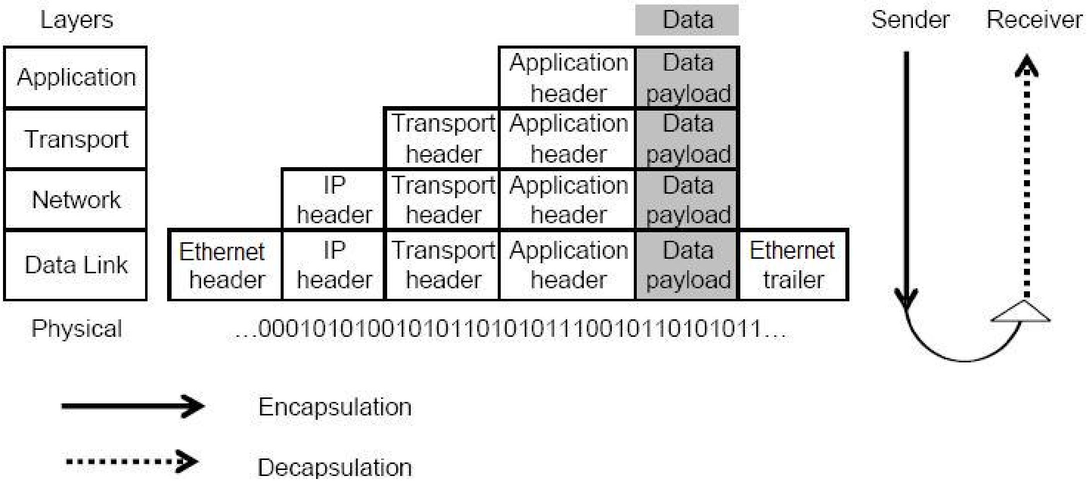


Figure 2.3: **Encapsulation and decapsulation**: special headers are added to packets when they pass through the layers, which identify the certain layers

2.4.3 Other divisions of networks

According to the area they cover networks are divided into several categories, typically:

- Personal Area Networks (PAN), a network connecting devices like telephones or Personal digital assistants(PDA). The devices are located close, a couple of meters, to one individual,
- Local Area Networks (LAN), cover larger areas than PANs, a few kilometres wide. These networks connect devices in an office, large building, school campus, etc,
- Metropolitan Area Networks (MAN), cover larger areas than LANs, as the name indicates. They are used to connect several LANs located in different sites of a large area,
- Wide Area Networks (WAN), is a network that covers a much larger area than PANs, LANs, or MAns. The best example of WANs is Internet. It connects many local or regional networks located in several areas around the globe,

2.5 Routing

As previously explained in section 2.4.1, in order to be able to exchange packets with a distant location, packets should be addressable. The addresses the packets include are the local and the destination addresses. Using this information, intelligent devices such as routers, find the best available path to send a packet from the source to destination. This process is called routing.

The node which carries out the routing contains a table according to which routing is done. This routing table holds the information needed to find the destination address, and what is the next node the packet should pass, to reach that final destination. That next node to be traversed is often referred to as the *next hop address*. The routing tables can be edited manually. Such routing tables are static. Instead, the routing to adapt to the modern network communications needs should be dynamic. This is where routing protocols help.

2.5.1 Routing protocols

The devices involved in a packet exchange should agree in advance about the "language" they will "talk" to each other. This set of rules and regulations which is used to carry out the routing process is called a routing protocol. Routing protocol is responsible for finding the path to route a packet to the destination address. A packet typically passes through several hops until reaching the desired location. Depending on how the route is calculated in the routing table, there exist two routing protocols

- **Distance vector** routing protocol, (often found named according to the developers Bellman-Ford and Ford-Fulkerson) is a protocol which is useful for small networks. Its routing algorithm uses hop count to calculate the routing table. This means that each router builds a table which shows the best known distance to any destination in the network, and how to get that destination. This information is exchanged with neighbouring routers.

- **Link-state** routing protocols are more effective in large networks. As this thesis deals with IS-IS link-state routing protocol only, then it is explained below.

2.5.2 IS-IS Routing protocol

This section will clarify some parts of the IS-IS routing protocol, which are necessary to know to understand this thesis. For more detailed information on the protocol refer to RFC 1195 [18], 1142 [29]. Before any traffic is exchanged between nodes, nodes should be physically connected. Next step is adjacency establishment. IS-IS does this by exchanging Intermediate-System Hello (ISH) packets, Link state packets (LSP) and Sequence Number Packets (SNP). Hello packets are used to initialize and keep adjacencies between the adjacent routers. LSPs are used to exchange information related to the state of links between routers [18]. To avoid receiving old information on link states, SNPs ensure that routers have the common view on which are recent PDUs. Using these types of packets every intermediate system (router) has a picture of the network, including all links and routers, and costs related to it. Only then traffic is routed between any two nodes along the minimum cost path which is computed using Dijkstras shortest path forwarding (SPF) algorithm [20]. A forwarding table is then constructed, associating an address prefix with the next-hop link [30]. The forwarding, i.e. routing tables are exchanged between all nodes in the network. In case any of the link's states changes, the routing tables are recalculated exchanging packets mentioned above.

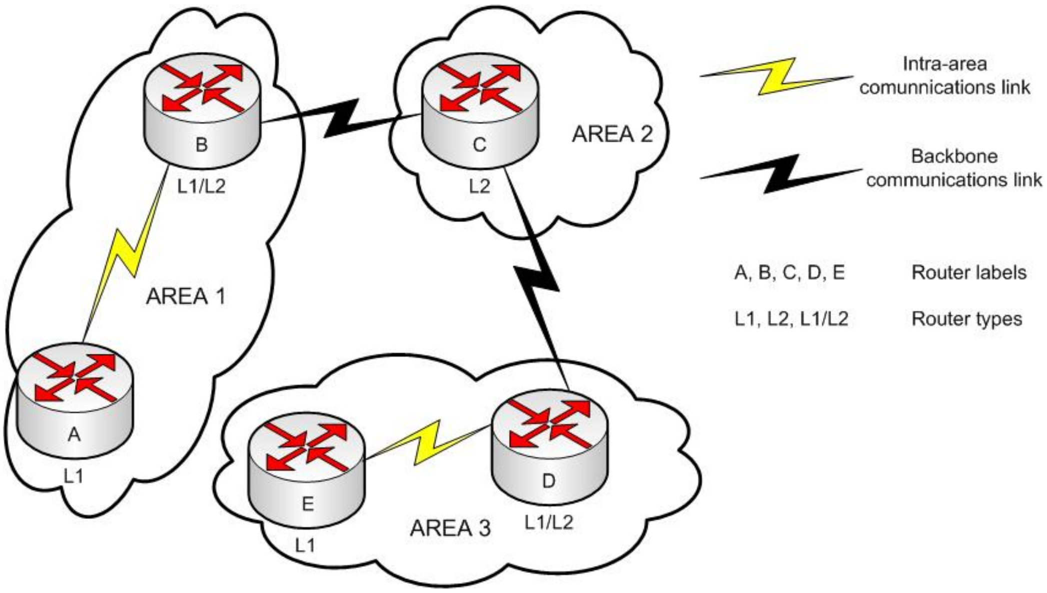


Figure 2.4: **IS-IS Areas and router types** - Areas identify routing domains, isolated from the rest. Instead they use border routers to reach other areas. Routers are labelled, where L2 and L1/L2 routers constitute the backbone of the network. L1 routers are used for distribution in the local networks.

Intermediate System (IS) is the ISO term for routers. As explained by the RFC 1195 [18] and RFC 1142 [29] Integrated IS-IS routing protocol is a dynamical link-state protocol, where the routing is done in a two-level hierarchical fashion. Packets forwarded by the protocol are transmitted directly to the underlying layer without any packet encapsulation. Using a hierarchical design with areas IS-IS provides the ability to hide instabilities within a problematic region from the rest of the network [31]. A routing domain, as a part of a network under the same administration authority, is partitioned into so-called *areas*.

Level 1(L1) routers know only about all routers and end systems their area. L1 routers do not know about the destinations outside of their area. Instead, L1 routers forward all traffic for destinations outside of their area to a level 2(L2) router in their area. L2 routers know the level 2 network topology, and which addresses are reachable through each level 2 router. Only L2 routers can exchange data packets or routing information directly with external routers located outside of the routing domains. In the boundary between L1 and L2 routers are L1/2 routers, which act as the gateway for L1 routers to external networks. L1/L2 and L2 routers constitute the so called backbone of the network, which is the core of the network. This is also the focus of this thesis.

Chapter 3

Related work

Stability of systems is closely related to the availability of services, i.e. high link uptime percentage. A system that has high availability over a period of time is said to be stable over that time span. Diot et al [26] have observed the availability of services in IP networks, and found out that not all link failures affect customer service. Besides defining some of the factors that affect service availability, they made the first step in defining a metric which quantitatively approximates service availability as a metric. This is a condition that typical SLAs fail to meet. SLAs guarantee only port availability, which is deals with a single component only, not the overall service availability.

The condition of network links is one of the main factors affecting the stability of IP network backbones. The duration of failures of network components could show indications about the source of errors as observed by Iannaccone et al [6]. They observed that when connectivity in a link of an IP network is lost for more than one hour, the failure could have originated from optical fiber restoration, or physical damages. Should the duration of a failure be less than one hour then these could be router or routing protocol problems. They were not able to monitor the failures caused by IS-IS routing protocol updates.

Markopulou et al [7], on the other hand, analysed the IS-IS data set collected during seven months in the Sprint's network IP backbone. Sprint is a large US based telecommunications company, that was the first to produce the world's only all-digital, fiber-optic network [1]. It is generally agreed that IP link failures happen as a result of several unrelated events at or below the IP layer. In their research they found that 20% of failures observed in their observed network belong to planned maintenance activities. Out of the rest of failures, 70% of failures which occur outside maintenance activities, affect a single link at a time. This finding is important to the analysis of the method in Section 5.

Failures caused by network routing problems are common. There have been many attempts to improve this situation. Nelakuditi et al [3] in their research have introduced a solution, to improve the failure resiliency, and at the same time not harm the stability of the network. Their solution includes a local rerouting approach, or as they call it failure insensitive routing(FIR). This is an attempt to reduce the link-state global

[1] http://www.sprintlink.net

updates, once link failure occurs, and instead have only rerouting calculations occurring locally. Once a link fails, the adjacent node is not notified of that failure. Instead the adjacency infers the failure, in case that node gets a message from an unusual interface, which according to SPF algorithm should not. In order to avoid routing loops FIR does local rerouting based on the incoming interface. Forwarding tables are calculated similarly to the conventional forwarding tables, but the main difference here is the identification of interfaces as ″usual″ and ″unusual″ to each node. According to the authors FIR is ″feasible, reliable, and stable″ and it reduces communication overhead.

Choudhury et al [32] look at the stability of the network in a rather different aspect. They investigate how data networks can be recovered fast by tuning the routing protocol configuration parametres. First they focus on the fast restoration under failure conditions, and then on improving network scalability and stability. In fast restoration phase, several parametres are changed, like the *Hello interval* or *SPF computation delay*. It is suggested to carefully tune these parametres, as they would affect the overall stability of the network. Additionally, LSA storm size, which is a large number of LSA updates, affects the network stability significantly. To not have important routing protocol packets queued with the rest of the data packets, Choudhury et al [32] suggest to mark the routing packets differently, using the optional fields that are unused in IP packets.. This way they would be prioritized, and network could recover faster than if queued with the rest of packets. Taking the measurements using this technique has shown improved results in the stability of the network they observed.

Shortest path from source to destination are calculated using the information on links of the network. This information, typically link weights, can be tuned by network operators manually. In an attempt to optimize the IS-IS routing protocol Fortz et al [21,22,33] provide optimization algorithms, to identify satisfactory protocol weight settings. They propose to make as few changes on the weights on links as possible. In this way congestion and link overload would be avoided. Link weights values usually reflect the inverse of link capacity. Changing weights on links is disruptive for a network, since that information has to be flooded across the network. Additionally, they advocate ways of engineering the traffic in IP routing protocols. Optimization is difficult, as traffic volumes are different over time, and unexpected failures can cause changes to the network topology [34].

The large amount of link-state updates flooded around the network is one of the main overheads created by the link-state routing protocols. In an attempt to reduce this overhead, Miyamura et al [35] observe ways how the link-state routing protocol scalability could be improved. They propose a way of reducing the overhead of routing protocol transactions. The algorithm they suggest limits the amount of neighbours that receive link-state updates. As their simulation showed, this way the network overhead is limited, and more reliable flooding is provided. A similar attempt to improve the fault-tolerance of the link-state routing protocol was done by Wu et al [36], by introducing a shortest restoration path for each uni-directional link fault.

Basu et al [37] take into consideration three factors when analysing network stability: the network convergence time, the routing load on processors, and the number of route flaps caused by failures. During their experiments using Open Shortest-path

First (OSPF) routing protocol, which is also a link-state protocol similar to IS-IS, they observed that concurrent failures cause major potential problems in stability of networks. Instabilities could be signs of overloaded processors, wasted router memory, high consumption of link bandwidth, and large amount of route flaps, i.e. frequent and fast changes reported by routing updates.

Improved convergence time for interior gateway protocols is one of the most important areas stability can be changed. The convergence time could sometimes be higher than a minute, but should the parametres be tuned well, they could give much better performance and higher availability. Francois et al [15] have tested changing several parametres in the routing protocol of large ISP networks, and ways how the convergence time could be decreased. They characterize the factors that affect convergence time in the following way:

$$\text{Convergence time} = D + O + F + SPT + RIB + DD$$

Where

- D is the failure detection time,
- O, is the LSP origination time, describing the new topology, once failure takes place,
- F, is the flooding time from the node detecting a failure,
- SPT, shortest path tree computation time
- RIB, is the routing table (Routing Information Base) update time
- DD , is the updates distribution delay

During their measurements and simulations they found out that if parametres are tuned carefully the results would be satisfactory. Some of the tuning they performed is explained in Section 6.4, using which they achieved sub-second convergence times, without any compromise on stability.

Chapter 4

Theory

> *"He who loves practice without theory is like the sailor who boards ship without a rudder and compass and never knows where he may cast."* - Leonardo Da Vinci (1452-1519)

As described in Section 2.1, modelling is needed to explain the stability related phenomena. Theoretical background is needed to explain the model or any scientific observation. The model proposed is a simplification of a real-life scenario, as it is impossible to include all factors that affect stability in networks. So the model is a mere approximation to the most likely reasons causing the phenomena observed and measured in the network. The problematic issues related to network stability are complex. Besides the diverse and numerous factors that cause the network instability, the spread of instability is a very important characteristic of analysing system state. Creating a hierarchy of elements in the network could be helpful helps in analyzing network elements, and ways to observe its stability. The hierarchy should reflect the importance of those elements in comparison to the rest. Additionally, the identification of important nodes shows where to focus more attention.

4.1 Theoretical principles used

4.1.1 Failure analysis

The importance of nodes and their connectivity is dependent on the failures in network links. The change in the state of the links is proportional to the change in ranking values and connectivity. Factors that cause network failures could originate from any of the TCP/IP or OSI communication layers, as mentioned in section 2.4.1. This shows how wide is the scope of failure origins: from a faulty user application, routing protocol addressing problems, physical addressing problems, to physical damages to links of nodes. Previous research mentioned in Section 3 showed that link failure durations of less than one hour indicate routing protocol problems. In the observed case-study network data logs provided by UNINETT for the period January-December 2005, these failures accounted for 95% of the reported failures. End-users would notice the degradation of the performance in several ways. When a link/router fails, traffic is rerouted via alternate paths, and may congest links along those backup paths [38]. Primary and backup IP-level paths may differ by tens of milliseconds. It has been shown that this is the major source of jitter in IP backbones [7].

By observing the frequency and duration of link and node failures occurring in an IP backbone, one can know how often there is need for rerouting, and disruptions in service delivery due to routing protocol convergence. This is characterized by the term Mean-time-to-failure (MTTF) or mean-time-before-failure (MTBF) which is the time between the end of one failure and the start of the next [7, 39]. Calculating the availability, using the DPM is another indication of how often failures occur in the network.

4.1.2 The centrality principle

The main element of the proposed stability monitoring method relies on the measurement of the centrality of nodes in the network. Network nodes can be ranked using Eigenvector Centrality calculation. Centrality is defined using the following formulas [2]:

$$v_i \propto \sum_{j=N(i)} v_j \tag{4.1}$$

which can be also written as:

$$v_i \propto \sum_{j=N(i)} A_{ij} v_j \tag{4.2}$$

which rewritten becomes in the form:

$$A\vec{v} = \lambda \vec{v} \tag{4.3}$$

where N(i) is the number of nodes, v_i is the vector for the importance ranking, and A is the adjacency matrix, a table of values reflecting the adjacency of nodes, i.e. what neighbouring, or adjacent, nodes are connected to other nodes of the network, and λ is the eigenvalue of that matrix.

$$\begin{array}{rc} & \text{columns i} \\ A = & \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & \cdots & a_{n,n} \end{pmatrix} \\ \text{rows j} & \end{array}$$

where i=1,2,...,n, and j=1,2,...,n, and they represent the position, columns and rows, of the element in the matrix. The adjacency matrix used here is a square matrix, i.e. the number of rows and columns is the same, $A_{n \times n}$. The elements of the matrix $a_{i,j}$, where $i = j$, are called elements of the diagonal of the matrix. In the matrices used in the calculations below matrix values are symmetric in relation to the diagonal. That means that for any i, j, $a_{i,j} = a_{j,i}$, for e.g. $a_{3,4} = a_{4,3}$. This is explained with the fact that the two nodes i and j are connected to each other. For our calculations the range of values for matrix elements are: $0 \leq a_{i,j} \leq 1$. If the value of $a_{i,j} = 0$, this means that nodes in those positions are not connected to each other directly. If nodes are connected then the value is larger than zero.

4.1.3 Node ranking

Eigenvector centrality is a calculation which uses the adjacency matrix to find central nodes in the network [2]. A square matrix has as many eigenvectors and corresponding eigenvalues as the matrix dimensions. The so-called principle eigenvector is what is needed in this calculation. The principle eigenvector is identified by searching for the highest eigenvalue from the calculation in equation 4.3.

Using the equation 4.3, n (as is the dimension of the matrix) eigenvectors and eigenvalues are obtained. We are interested in the principle eigenvector. The principle eigenvector is the one with the highest eigenvalue. After that vector is located, it is sorted by ranking value. This way the most important nodes, i.e. nodes with highest ranking value, are found on top of the list.

Once nodes are ranked, one can identify what harm would be caused if certain nodes were to be removed, or damaged. Nodes which can cause an overall instability when removed are ranked as top-most nodes. The effectiveness of the proposed method is that it identifies these important nodes. Once identified, one should apply more stability tuning to those nodes, to prevent the instability to spread out across the whole network. This is an important element in developing a model for network stability. This ranked list of nodes should be inspected against the levels of instability as described in the scale in Table 5.1.

A similar principle eigenvector calculation has previously been used to measure aspects of social networks like popularity [40], other aspects of social networks [41], Google[1] to rank their pages to be searched, using an algorithm called Pagerank [42], for modelling probable behaviour of networks, including eigenvector methods [12, 16], etc.

4.1.4 The network connectivity

The calculation of the overall connectivity, χ, of a network N can help in finding how possible it is for a message to cross directly between any two nodes in the network [2]:

[1] http://www.google.com

$$\chi = \frac{1}{N(N-1)}\vec{h}^T \times A_{n\times n} \times \vec{h} \tag{4.4}$$

where $\vec{h}^T$ is the transpose vector of $\vec{h}$ are vectors with dimensions which correspond to the number of rows, n, of adjacency matrix, A_{nxn}, of the network. $\vec{h}^T = (p_1, p_2, \ldots, p_N)$, where $0 \leq p_i \leq 1$, and p_i is the probability that node i is available. If the probabilities are 1, the hosts are said to be reliable, otherwise they are partially or not reliable. The *connectivity*, χ, of a network graph G, is the probability (averaged over all pairs of nodes) that a message can be passed directly between any two nodes. The values for connectivity are $0 \leq\chi \leq 1$. If the connectivity value is 1 then all nodes in that network are directly connected to other nodes. If nodes are not connected the value of χ is minimal, 0. Change in connectivity is an indication of change in the state of links and nodes in a network, thus is an essential part of the stability monitoring method.

4.2 A more efficient stability monitoring model

Connectivity and node ranking calculations using only the method mentioned in the previous section are not helpful if we need to observe stability of a system. Instead, to make the calculation usable for observing stability, this thesis proposes a modification of the method. In the example graph shown in Figure 6.1 there is no detailed information to characterize the role of the node in the network. Because node B is the most connected node, it might not be the most important. Other factors should be taken into consideration. Information such as link metrics, node availability, packet loss rate, etc. could give a better insight into ranking the nodes in a more fair way. The same graph example will be used to illustrate this following reasoning.

Now instead of calculating connectivity and ranking using only fixed adjacency matrix values, $a_{i,j}$, which show whether the nodes are connected to a respective neighbour, a different value, $k_{i,j}$, will be inserted in the scaled adjacency matrix, A. The scaled adjacency matrix, which includes the parameter $k_{i,j}$ looks as follows:

$$A_{ij} \rightarrow k_{i,j} = \frac{a_{i,j}}{M \times D \times L} A_{ij} \tag{4.5}$$

where:

- $a_{i,j}$ is the adjacency matrix value, in the i and j coordinate, with values $0 \leq a_{i,j} \leq 1$, 0 if node i and j are not connected, else it is 1,
- M is the metrics, which is manually assigned value, and reflects the inverse of the capacity, and delay in the link connecting the respective nodes relative to the regional links (see Figure 6.8),
- D is the downtime percentage of the link connecting those nodes,

- L is the percentage of packet loss in that link connecting those nodes.

Node downtime or node packet loss is irrelevant to include in the calculation, as when links fail completely, or do not deliver packets, that means the same as saying interfaces in the nodes do not respond. A link is defined as an established connection between one interface of a node with another interface in another node.

The new principle eigenvector derived from the scaled adjacency matrix is no longer the same as the pure adjacency matrix. Nodes which are connected through more reliable links and of higher capacities get a higher importance than those nodes which are unreliable and of low capacities. This introduces a rather more effective importance assignment method to nodes than the node ranking method using pure adjacency matrix. Therefore, the parametre k is considered an effective expected adjacency. These analysis, and other findings will be shown later in section 6, where clearly will be seen that unreliability in the neighbours will affect the node itself.

Chapter 5

Methodology

> *"True stability results when presumed order and presumed disorder are balanced. A truly stable system expects the unexpected, is prepared to be disrupted, waits to be transformed"* - Thomas E. Robbins, an American writer.

To explain the proposed model of analyzing network stability, this chapter is divided into four parts. The first part explains the experimental analysis, assumptions, and the way data are obtained, to be useful for network stability monitoring. The second section of this chapter explains the algorithm to follow for analysis of the proposed network monitoring model using the data obtained in the first section. The analysis procedure is explained, and visualized through a diagram 5.1. More detailed findings and results of analysis on the network of the case-study are explained in Chapter 6. The last part of this chapter lays some hypothesis which are expected to be verified with the results obtained from the experiments.

5.1 Experiment analysis

Using eigenvector centrality principle explained in section 4 nodes of the network are ranked and the connectivity value of the network is calculated. The first part of the experiment takes a simple topology as an example to illustrate how efficient the method is, and its ability to identify instability spreading across the network. Second part of the experiment takes the UNINETT case-study network, but hypothesizes link failures, to see how efficient this method is in a larger scale network, two main links of which are suffering various downtimes. The third part of the experiment analyses the case-study network, including real link downtimes caused by failures observed in February and March of 2006.

5.1.1 Obtaining data

Data used in the second and third part of the experiment are retrieved from UNINETT's publicly available logs[1]. The logs contain the duration of the failure of nodes and links. The format of the data logs for nodes agents is:

```
                  Agent downtime           Agent fail          Unknown   Total
Agent          #>8h >1h>10m<10m   All    Duration  Avail.    #  Duration  Avail.
---------------------------------------------------------------------------------

stolav32-gw      8   1   3  12   24   495:41:11 94.341%   0   0:00:00  94.34%
stranden3-gw     6   0   1   2    9   311:20:54 96.446%   0   0:00:00  96.45%
```

and the format of the logs for the link failures is:

```
                      Link downtime           Link fail          Unknown   Total
Link             #>8h >1h>10m<10m   All    Duration  Avail.    #  Duration  Avail.
---------------------------------------------------------------------------------

sarpsborg-gw(21)    2   0   8   7   17    45:04:35 93.292%   0   0:00:00  93.29%
trd-gw(12)          1   3   1   2    7    17:01:49 97.466%   0   0:00:00  97.47%
```

Where agent or link downtime durations are the length of a duration, and the number in that column the number of occurrences of that kind of downtime per period of time. Figure 6.11 and Figure 6.12 are created using these logs. Furthermore, the centrality of a certain node is calculated using the percentages of total availability of links, and the values are inserted in the value of the equation 4.5 as a part of the adjacency matrix.

The information on the metrics is obtained from the assigned values as shown in Figure 6.1, and those values are inserted in equation 4.4 and equation 4.3.

5.1.2 Assumptions

Modelling a system behaviour means that the system analyzed is short of some parametres, which could affect the system. Hence, the experiment and the model itself is based on several assumptions. One of the main assumptions is that links do not fail simultaneously. Even if that possibility is small, some analysis is shown should that scenario take place, to illustrate interesting results the method reveals. It is assumed that overall link uptimes are between 99.9% and 55%. The hypothesis here is that with the uptime of links decreasing, some high-capacity and high-rank nodes will suffer decrease linearly in the ranking value. Instead, lower-capacity and low-rank nodes will become more important, and will suffer overload. The case study network, UNINETT has logs gathered during longer periods, around 4 years, but the assumption is that updates and upgrading have taken place in the network topology, so it is irrelevant to deduce future system behaviours based on old logs. Instead recent logs on the state of the network topology are analyzed.

[1] http://drift.uninett.no/downs/ last time accessed in May 20, 2006

5.2 The algorithm of the stability monitoring model

The complex task stability monitoring analysis is simplified and encompassed in two steps (see Figure 5.1). Both separate steps start by calculating the parametres according to the wanted state, the SLA level of provision. SLA compliant system snapshot is then compared to dynamically changed states. This snapshot should reflect what the system policy considers to be an acceptable ranking of nodes by their importance. The acceptable level is related to the minimum service provision secured by an SLA agreement. The SLA compliant ranking is done using the manually assigned metrics, and adjacencies of nodes and links in the network. Snapshot does not consider any changes in the level provision, but assume the maximum possible level of service is provided.

First step of the analysis is about comparing the connectivity value according to the SLA, and comparing it to the new connectivity value calculated when taking link downtimes into consideration. Should the value of connectivity be within the accepted level, i.e. between maximum theoretical value and SLA value, then the system is within the stable threshold. With SLA value we mean the change in ranking value that signifies a threat to stability, because this indicates where vulnerabilities can have a big effect on connectivity. If it is below the SLA value of connectivity, then instability is suspected. This leads to the second step of the analysis. In the second step values of node rankings compliant to SLAs are compared to the new ranking values which are calculated using the link downtimes, and metrics. If the respective nodes ranking value is changed beyond the SLA level, then those nodes should be analysed further. If the ranking value of a node is higher than in the previous SLA snapshot, then that node has become overloaded, or the other way round. Highly overloaded node could be a source of instability to the rest of the nodes.

System state changes continuously, due to the interaction with external and internal factors surrounding it. The model observes changes that are noticed in the snapshot ranking of nodes. Changes are manifested with downtimes in links, packet loss, etc. Considering parametres such as link downtime, and packet loss, the proposed stability model provides the ability to see deviations from the statically calculated snapshot ranking. A scale should be defined, to specify what deviations from the ranking should be considered low, and what high instability rates. As shown briefly in Section 2, typical SLAs try to enforce a 99.9% port availability. Port availability refers to the uptime of a network element [7]. In our case, the interfaces which establish links between nodes should guarantee that level, to comply with the SLA. The stability observation method proposed will define a maximum and minimum value of ranks for each node in the network. Should the level be lower than the SLA values, then clearly there is instability in that node. That instability could affect the surrounding areas, as will be shown in the Section 6.

5.2.1 The instability scale

An instability scale is needed, in order to make it clear what change in the ranking values should be considered an urgent issue, and what change is within the tolera-

Instability Level Suspected	δ (SLA vs current) in %
Low	$0 \leq \delta \leq 25$
Medium	$25 \leq \delta \leq 50$
High	$50 \leq \delta$

Table 5.1: **Scale** - A preliminary scale is given to compare the change in the ranking value, and to signal levels of instability according to that. These values should be tuned to fit the topology which is analyzed

ble threshold. The scale given in this example is preliminary, and should be taken with a certain level of uncertainty. The scale comprises of three levels describing the suspected instability, low, medium or high.

The urgency of repair should be prioritized according to the ranking of nodes by importance as shown in section 4.1.3. In addition to the diagram shown in Figure 5.1, to compare the level of instability, and the urgency of repair, the diagram in Figure 5.2 is used.

5.2.2 Monitoring routing stability

Previous parts of the methodology observed the failures in the network, and proposed a method to observe the ranking of values. Observing the change in ranking shows what areas one needs to focus more, to improve the stability of routing. Previous research in IP backbone network dynamics showed that a considerable number of failures in the network are due to router and routing protocol problems [7,15]. Typical routing problems are the routing loops. Though the IS-IS link state routing protocol deals with routing loops, some mistakes are inevitable. Some network protocol tuning can help prevent this problem in the future (see Section 6.4).

5.2.3 Uncertainty in the model

As any scientific measurement, the suggested method may not give precise indications of the stability in networks. As it is almost impossible to encompass all the factors affecting stability of networks, the proposed method is a mere approximation of its possible true factors. The calculations have been repeated several times to avoid personal errors added to the estimations. Random errors are unavoidable, but the assumption is that they even out while repetitions of measurements are carried out.

Future work on the model includes quantifying the uncertainty, and ways to tune the ranking values to reflect more accurately the importance of nodes in a real network topology.

5.3 Hypothesis

On grounds of the theoretical model mentioned above, the results from the experiments described in the next section 6 are expected to confirm the hypotheses posed below. As discussed earlier in the chapter, using the value of connectivity reflects the state of the links and nodes in the network, and changes respectively. Hence, the first hypothesis:

Hypothesis 5.1 *A decreased value of the overall of a network connectivity value, χ, is a potential sign of instability in the network.*

The newly introduced adjacency matrix is expected to picture better the real ranking of nodes in the network, as it includes more information on the topology and reliability of parts of the network:

Hypothesis 5.2 *The ranking of nodes by importance, using principle eigenvector with the k effective expected adjacency matrix is a more accurate ranking method than using the pure adjacency matrix.*

The dynamics of change in the ranking values gives hints on how nodes are affected by the failures in parts of the network topology. From this assumption we derive the following two hypothesis:

Hypothesis 5.3 *If the value of the ranking value of a node decreases then that node is underloaded, and is not being utilized well. This results in surrounding nodes getting overloaded, and may experience packet loss.*

and,

Hypothesis 5.4 *If the value of the ranking value of a node increases then that node is overloaded, and is not being overutilized. If no action is taken this results in the responsiveness of the node, and packet loss is expected.*

In a worst case scenario, when nodes (and links as a result) fail simultaneously, and suffer the same rate of downtimes then the value of connectivity should reflect that change:

Hypothesis 5.5 *If the links of a network fail in a simultanous manner, and suffer the same amount of downtime, then the value of connectivity will change linearly.*

Hypothesis 5.6 *SLA value is the change in ranking value that signifies a threat to stability, because this indicates where vulnerabilities can have a big effect on connectivity. If it is below the SLA value of connectivity, then instability is suspected.*

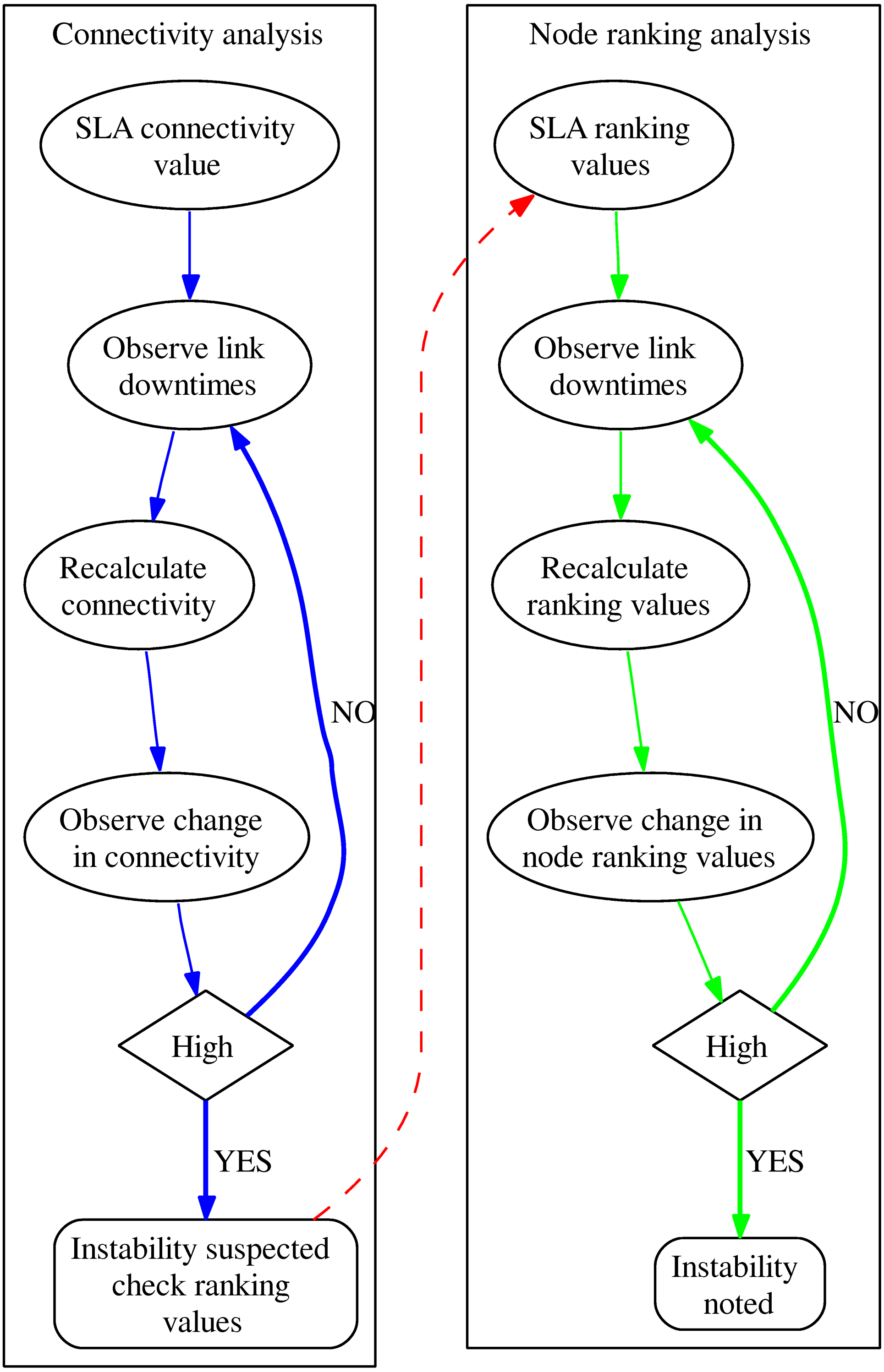


Figure 5.1: **Analysis procedure** - First connectivity value of the network is traced for changes, and if indications show that, the ranking values are analysed also. If significant changes in the ranking values are noted, then nodes which suffer from instabilities are shown.

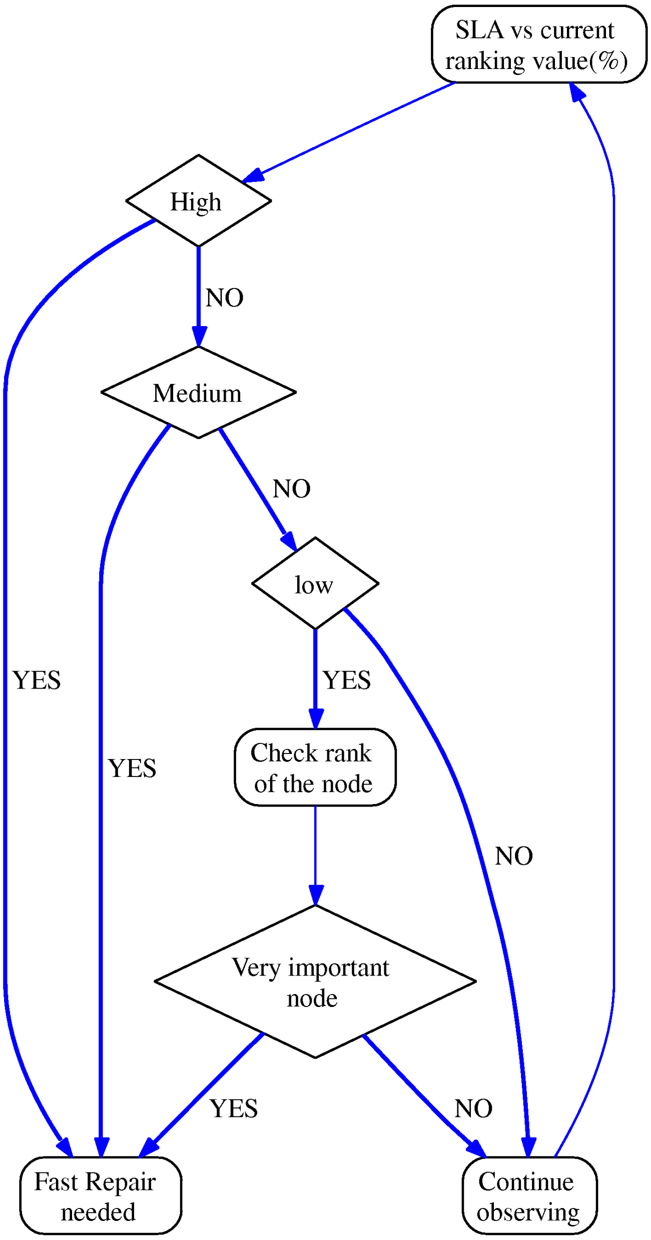


Figure 5.2: **Scale** - Before one alerts about instabilities, as shown in Figure 5.1, first the change in the ranking value is compared to the scale in Table 5.1, and levels of instability are assigned. The urgency for fast recovery is stated according to the instability level, and the importance of the node.

Chapter 6

Results and analysis

> *"No amount of experimentation can ever prove me right; a single experiment can prove me wrong."* - Albert Einstein

This chapter explains the results obtained during the application of the model explained in section 5 and whether the hypothesis stated there are proved to be correct. This chapter is divided into three sections. Each section is first analyzed using conventional tools and then the solution proposed in this thesis. The conventional tools are either the monitoring tools available today for network monitoring, or the node ranking method using a pure adjacency matrix. The first section of this chapter analyses how efficient is the model for a small-size network composed of five nodes connected by six links, as shown in Figure 6.1. The second section uses a hypothetical situation, when two main links of the case-study topology are simulated to fail for a certain period of time. This way the proposed method will be tested to check if it gives credible results. The third section, UNINETT network analysis, deals with data gathered from live running network, and observation of the reported failures. It first includes an analysis of failures observed in the UNINETT IP backbone, and then using that information uses the proposed stability monitoring model to analyse whether instability has taken place in the network. Last section includes tips on how the network stability can be improved, alternative protection that can be applied in networks, ways to improve the routing protocol convergence speed, etc.

As it can be seen from the Figure 6.13, the network backbone topology consists of regional redundant links around the main nodes. There are 63 bi-directional links in the network. Links consist of a mixture of optical *packet over SONET/SDH* (POS) links from 155 Mbit to 10 Gbit links and some GigabitEthernet point to point links. Some of the links marked with bold lines have a capacity of 2488000 kbit/s, some 155000 kbit/s. The rest of links, mainly regional backup lines, vary in capacities, down to 32000 kbit/s. There are 46 Cisco routers in the network backbone, from various series, 7200, 7500, 7350ME, 10700 to 12000. It is important to stress the fact that in this analysis one important fact is left out: the three top-most nodes carry the traffic to Sweden, through a high capacity (9953000 kbit/s) link. Having this information one can by inspection deduce that the three top-most nodes will be in the Oslo region. As will be shown in the next sections, the method proposed identifies those nodes as well, even

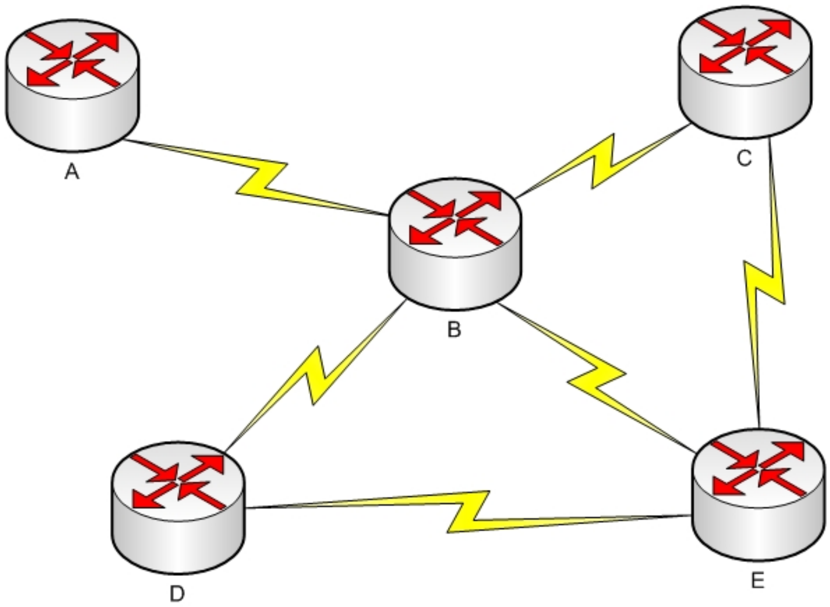


Figure 6.1: **Simple topology** - The network is composed of nodes and links, but there is no information on any link metrics. Hence, the assumption is that all links have the same capacities, and nodes are ranked accordingly.

if this important link information is not considered in the calculations.

6.1 Application of the model in a simple network

The following example will show how a simple network can be ranked, using calculation of the eigenvector centrality of the network's adjacency matrix. The calculations are done using scripts, which perform the calculations using Octave, a high-level language, primarily intended for numerical computations[1]. Further explanation on the scripts are shown in Appendix A.1. The network topology looks like shown in Figure 6.1:

6.1.1 Conventional method of ranking network nodes

First one should identify the adjacency matrix of the given network topology. An adjacency matrix shows the existing connections between nodes. In this example a pure adjacency matrix is used. If two nodes are connected between each other, the corresponding matrix value will be one ″1″, else it will be marked with a zero ″0″. The following adjacency matrix assumes that the links are bi-directional, i.e. traffic flows in both directions. Therefore all matrices used in the next calculations have ″0″ as values in their diagonal elements, and are mirrored around that diagonal.

[1] http://www.octave.org last time accessed in May 20, 2006

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

This adjacency matrix applied to the equation 4.3 we obtain the following Eigenvalues:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} Eigenvalue \\ \begin{pmatrix} 2.686 \\ -1.749 \\ -1.271 \\ 0.335 \\ 0.000 \end{pmatrix} \end{array}$$

and the following eigenvectors:

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0.217 & -0.370 & 0.315 & 0.846 & 0.000 \\ 0.583 & 0.648 & -0.401 & 0.283 & 0.000 \\ 0.412 & -0.458 & -0.283 & -0.200 & 0.707 \\ 0.412 & -0.458 & -0.283 & -0.200 & -0.707 \\ 0.524 & 0.153 & 0.761 & -0.351 & 0.000 \end{pmatrix} \end{array}$$

The most central nodes have the highest values in the eigenvalue table, so that value should be located first. Looking at the Eigenvalues we see that the highest value is 2.686, and is in position ″1″ in the table. This means that we should inspect the first column of the eigenvector table and find the highest value, which is 0.583, and corresponds to node labelled with letter ″B″. This means that node B is most important node, is the most connected, and has important neighbours connected to it. The same eigenvector column is sorted, and ranked according to the importance the ordering of nodes is as follows:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cc} Label & Value \end{array} \\ \begin{pmatrix} B & 0.583 \\ E & 0.524 \\ C & 0.412 \\ D & 0.412 \\ A & 0.217 \end{pmatrix} \end{array}$$

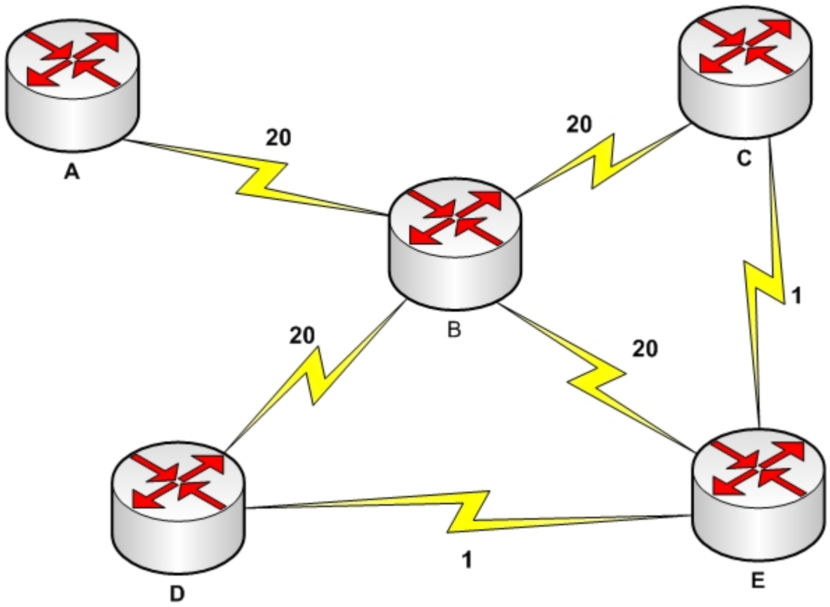


Figure 6.2: **Simple topology** This network is the same as Figure 6.1, but now there is more information about the links. The metrics on the edges show the inverse of the capacities of those links.

As we see node E is the second most important node, nodes C and D are equally important, and node A is the least important in comparison to the other nodes in the network.

6.1.2 Adapted model for monitoring stability

To prove the hypothesis 5.2 that the effective expected adjacency is a more accurate method we use the topology as shown in Figure 6.2 is used. As we can notice link metrics are specified in the topology and it is assumed that the rest of metrics such as downtime rate or packet loss rate is roughly 1%. Doing the same calculations as in the first example we get the following ranking:

$$\begin{array}{cc} & \begin{array}{cc} Label & Value \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{pmatrix} E & 0.706 \\ D & 0.499 \\ C & 0.499 \\ B & 0.060 \\ A & 0.002 \end{pmatrix} \end{array}$$

The ranking of nodes in the second example is quite different from the first. Node E now is far more important than node B which is ranked the fourth according to importance. This is due to the capacity of the links surrounding node B, which are low capacity links. As we can see the second way of calculating the eigenvector centrality reveals a very important fact. When more parametres inserted in the equation give a better distinction of what nodes are more important. If nodes are connected physically it does not mean that they are connected all the time. Instead the connectivity depends on the state of the links connecting them. It is a well-known fact that no link can provide 100% availability. Therefore, one gets a more accurate ranking if nodes are assigned values according to link capacities surrounding the node, the historical data on the availability of the node, as well as packet loss.

Connectivity also is calculated using the modified adjacency matrix values. This way the connectivity will also be based on the downtime duration of links and capacity of the link. Monitoring changes connectivity factor in time could also give insights on how stability of the network is changing.

6.1.3 Stability analysis using the proposed method

To see the ability of the method to trace stability the links we assume that the links uptime from node C to E is 70% and from D to E is 74%. (These link uptime percentages were randomly generated using web based tool [2]). This way, the hypothesis 5.1 will be verified. That will be proven true, if the connectivity decreases, due to the link failures taking place as described above.

Following the diagram as shown in Figure 5.1 first the connectivity value is calculated, assuming maximum link availability. The initial value for the network is χ_{h-SLA} = 0.0021252. When the downtimes were inserted in the adjacency matrix the new value of connectivity became χ_{hn-SLA} = 0.00026741. The hypothesis in 5.1 is correct, and this change in connectivity value is an indication that instability is taking place somewhere. Now the second step of the analysis measures change in the ranking value. After calculating the new ranking values, and comparing them to the SLA values, the results are shown in Figure 6.3 and Table 6.1.

Node name	SLA ranking	New ranking	Δ (Rank value)	Δ (Rank value) %
A	0.002117363	0.02464241	-0.244306737	1063.82%
B	0.60105715	0.61515971	0.01410256	2.35%
C	0.49925018	0.37795756	-0.12129262	-24.29%
D	0.49925018	0.41824709	-0.08100309	-16.22%
E	0.70560706	0.49302246	-0.2125846	-30.13%

Table 6.1: Change in node ranking values caused by added failures in links. The nodes the ranking values of which increase, they suffer overload, and might be a source of instability in the network.

6.1.4 Discussion

The results shown in the previous part identify that node E out of which links suffering downtimes originated becomes less utilized, and as a result packet loss might

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

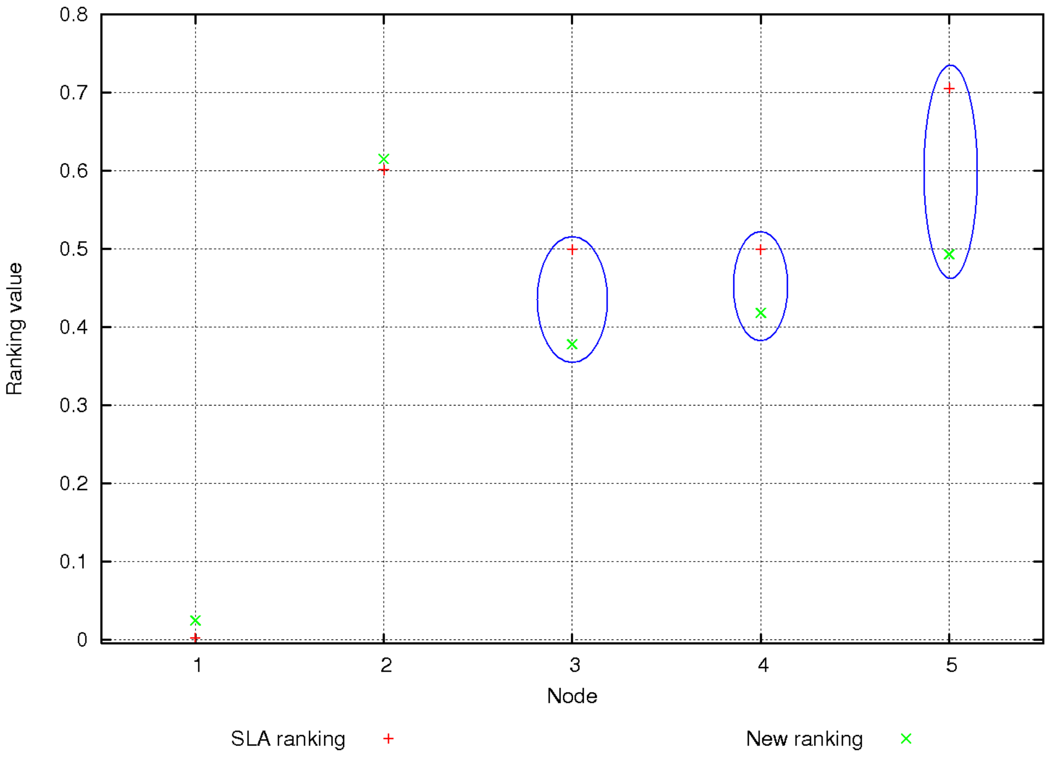


Figure 6.3: **Change in ranking**: The network shown in Figure 6.2 is analyzed, including information on link failures. The change in ranking values is due to link failures, where link C-E has uptime 70% and link D-E 74%

be taking place in that area. Node B which was the second most important in the first case, now becomes the most important node, and is overloaded, as its ranking value shows. As metrics on the links indicate, congestion could happen, as the links surrounding node B have lower capacity than nodes around E. These results clearly show that the method is efficient for small networks. Looking at Table 6.1 we see that the ranking value of the important node ”E” has decreased for more than 30% which according to the diagram 5.2 is a serious instability level. Additionally according to the hypothesis 5.3 nodes surrounding node ”E”, (nodes D and C), are experiencing packet loss.

6.2 Simulated failures in UNINETT backbone

Similar to the steps taken in the previous section, in this section the topology of the UNINETT backbone is used to analyse the proposed method. Obviously, conventional methods do not help in analysing stability of network nodes, so only the proposed method is used in this example.

The link failures are simulated, in order to observe changes in its IP backbone

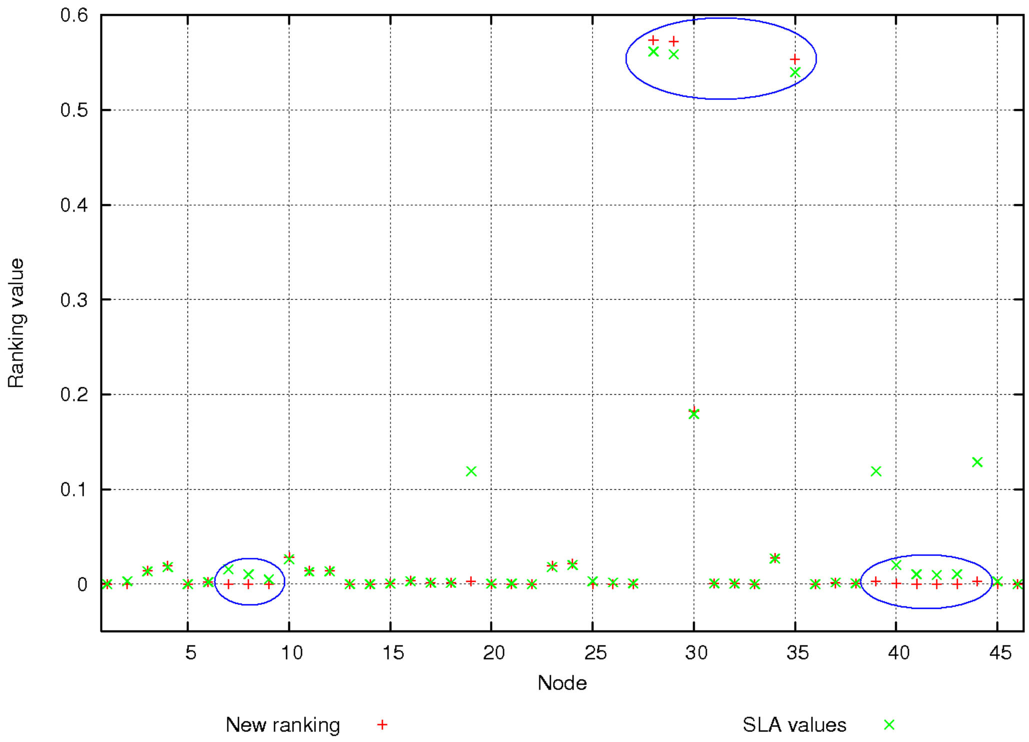


Figure 6.4: Observed change in the new measurement, in comparison to the SLA ranking values. The link between o1 and trd gets an uptime value 56%, and another important link uptime between o2 and brg becomes 85%. Node 19 (hb), 39 (tb), 44 (trd) values decrease, and node 28 (o1), 29 (o2) and 35 (sto) increase in rank value

stability. It is assumed that only two of the network links have suffered major failures, and the rest have been stable. Using the web tool to generate random values [3], the link failures of two main links originating from two most important nodes, ″o1″ and ″o2″ are randomly chosen. The link between ″o1″ and ″trd″ gets an uptime value 56%, and another important link uptime between ″o2″ and ″brg″ becomes 85%. After inserting the information on failures in the equation, then the value of connectivity is χ_h = 0.019553 and compared to χ_{SLA} = 0.019786 it is lower for a value 0.000233. According to the hypothesis 5.1 this means that instability may be taking place in the network. Now we have to reveal what region is suffering from instability.

In the second step, ranking values of the network are observed, to see indications of instability. Nodes are assigned numbers, and are ordered alphabetically as shown in Table B.1.

[3] http://www.random.org last time accessed in May 20, 2006

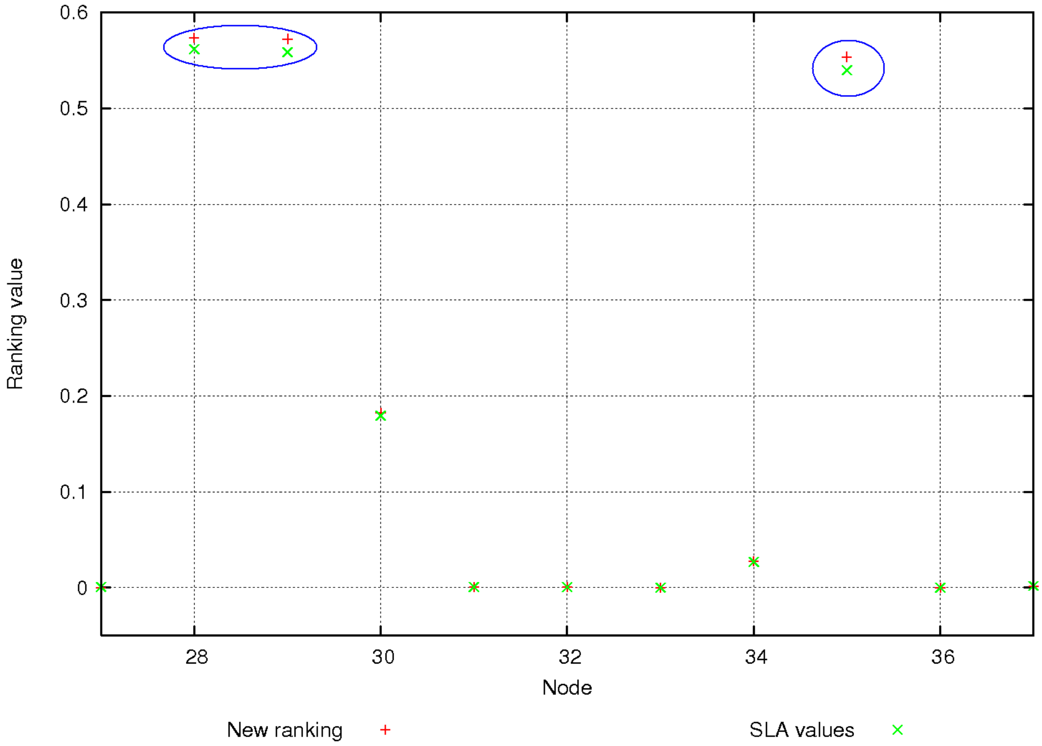


Figure 6.5: Zooming a part of Figure 6.4, we observed change in the new measurement, in comparison to the SLA ranking values. Nodes 28 (o1), 29 (o2) and 35 (sto) increase in rank value

6.2.1 Discussion

Both decreased and increased ranking values signal instabilities in the network. Looking at Figure 6.4 it can be seen that two most important nodes, ″o1″ and ″o2″ out of which main links originate and suffer failures get higher ranking values i.e. they get overloaded. This is explained with the fact that high capacity links are incapable of forwarding traffic, thus alternative lower link capacity links need to be employed. On the other hand, nodes ″hb″,″tb″, and ″trd″ get underloaded, because of failures in two main links, and this results in lower ranking values. Underloaded links bring inefficiency to the network, and risk that other lower capacity links will be congested. Node ″trd″ is responsible for forwarding traffic from the main link to the other two nodes (″hb″ and ″tb″), which in return now receive the traffic through alternative nodes.

The Table B.1 shows that several important nodes of the network suffer significant instabilities, which according to the scale shown in the other Table 5.1 are defined as high-level instabilities. As a result fast recovery is advised by the diagram in Figure 5.2, otherwise serious damage can be caused to the rest of the network parts.

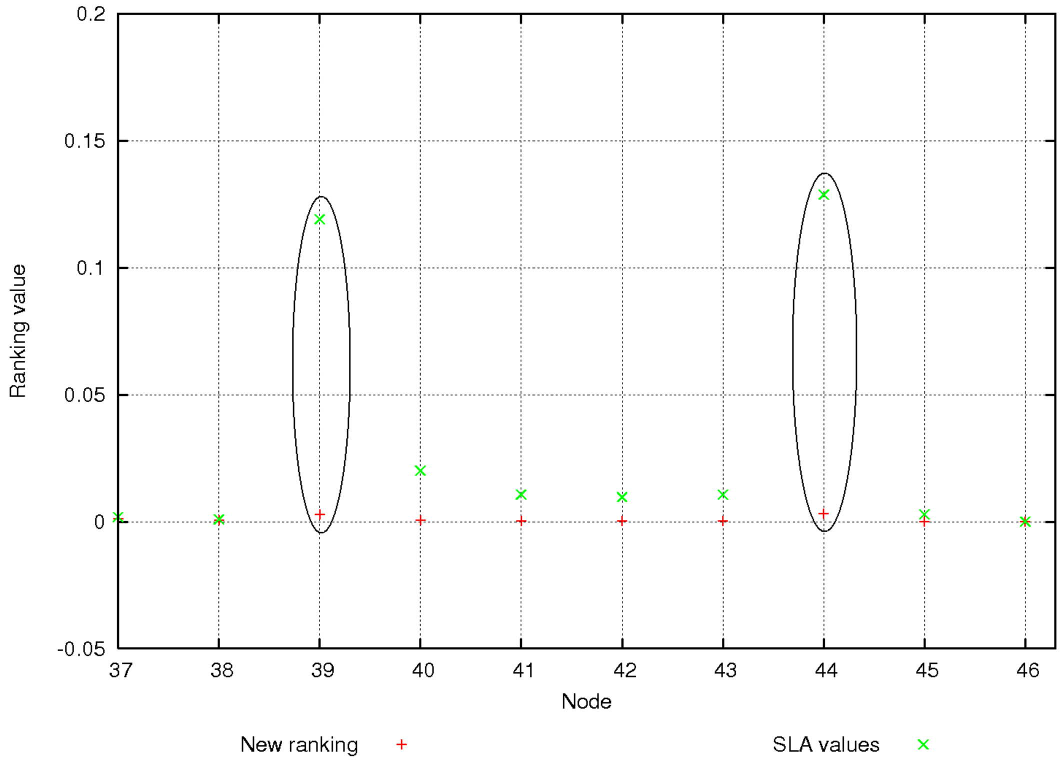


Figure 6.6: Zooming a part of Figure 6.4, we observed change in the new measurement, in comparison to the SLA ranking values. Nodes 39 (tb), 44 (trd) values decrease in rank value

6.3 Monitoring real failures of UNINETT backbone network

To get a better picture of the current situation in the UNINETT network, the length and other aspects of failures that occurred in the past are analyzed. Additionally, UNINETT network backbone topology is analyzed, including its connectivity and its most important areas. Such findings would then be suggested in tuning of the live network. UNINETT has configured a backbone with regional redundant links around main nodes (see Figure 6.8).

6.3.1 Network monitoring using conventional tools

In this subsection three conventional methods for monitoring network failures are presented: *rtanaly*, DPM, and retrieving failure information using simple network monitoring protocol (SNMP).

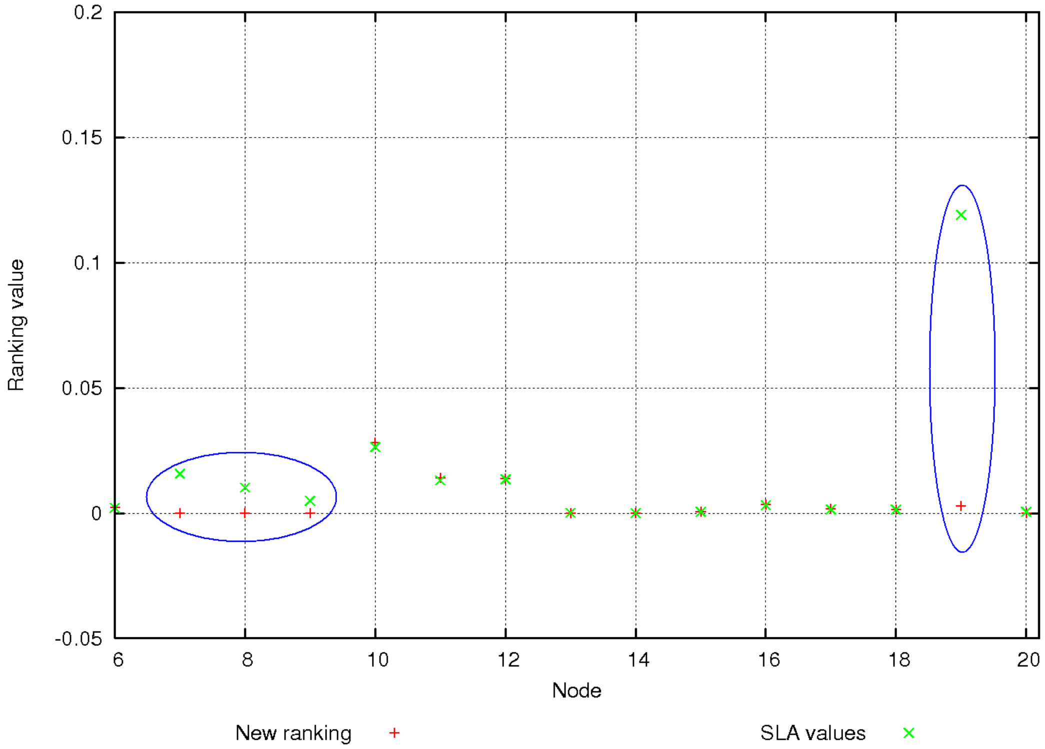


Figure 6.7: Zooming a part of Figure 6.4, we observed change in the new measurement, in comparison to the SLA ranking values. Nodes 7 (brg), 19 (hb) values decrease in rank value.

Monitoring routing protocol information transactions

In UNINETT, IS-IS routing protocol transactions are observed and analyzed, to identify the routing information changes throughout the network. The logs on the transactions are obtained using an IGP transactions logging tool, *rtanaly* [1]. Rtanaly is a tool which works in a client-server fashion. Several clients are distributed around the network, fetching information exchanged during routing protocol information exchanges. Rtanaly has the ability to investigate the changes on the link states between the distributed nodes. That information on rtanaly transactions is open to public, and can be easily retrieved [4]. A central node is configured to receive the information from the distributed logs, and displays the received information in a web browser, as it is shown in the Figure 6.9:

Observing the LSA exchange rate one can detect the size of the LSA storm, which affects the stability of the network. Rtanaly does not have the ability to signal instabilities, it merely observes the amount of LSA packets and what routes are changed during a certain period. The Link-state Advertisements (LSA) storm size, which is the large amount of LSA updates, could indicate instability in the routing process. UNINETT network backbone is analysed to trace sources of failures. This is done by

[4] http://drift.uninett.no see link named ”ISIS rutingstatistikk” last time accessed in May 20, 2006

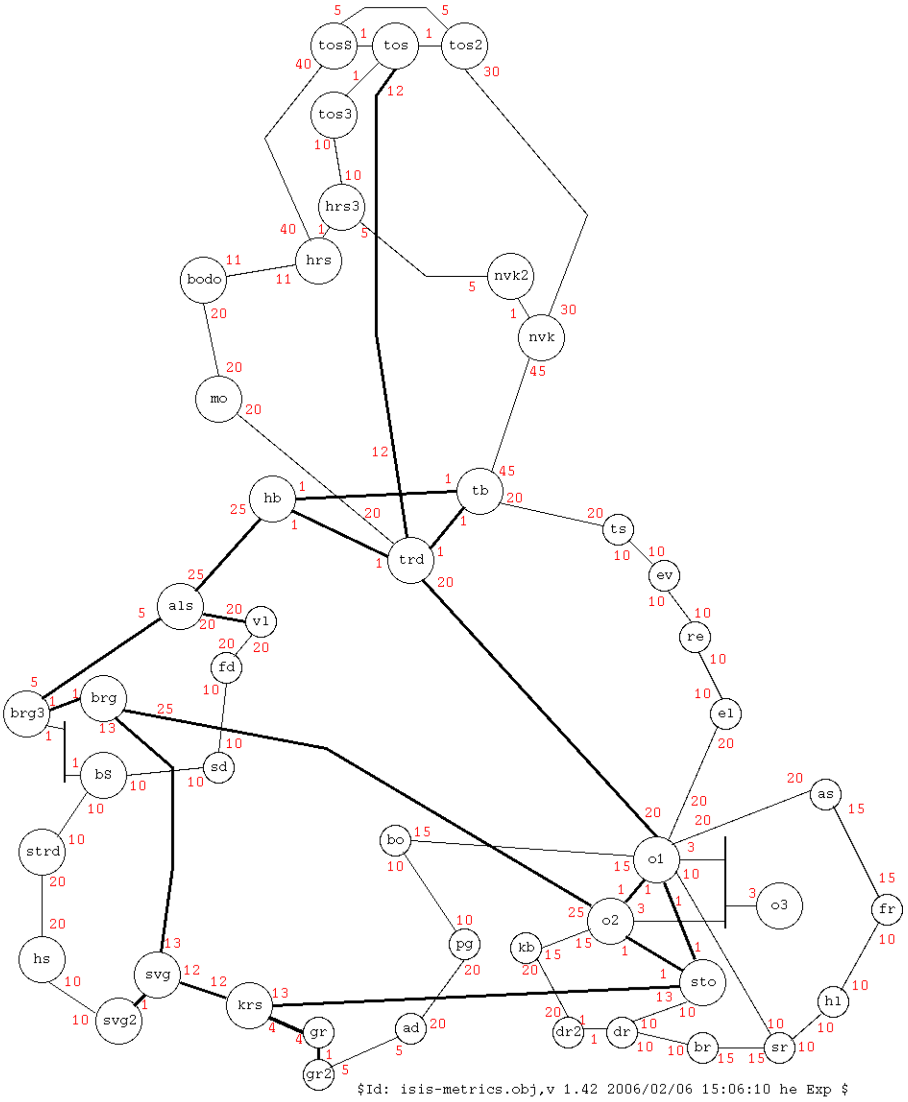


Figure 6.8: UNINETT IP Backbone nodes, with their regional redundant links, and with specified metrics in the links

observing the failure durations in the nodes and links of the network. As rtanaly reports, the rate of LSP changes this network has exchanged December 2004 to March 2006 [5] is as shown in Figure 6.10.

Monitoring individual device failures

UNINETT has gathered data to measure the duration of the downtimes in individual devices, caused by failures in links and nodes of the network shown in Figure 6.8. This is done using SNMP-based tools, which receive packets of information that describe

[5] http://drift.uninett.no/ ”ISIS rutingstatistikk” last accessed in May 20, 2006.

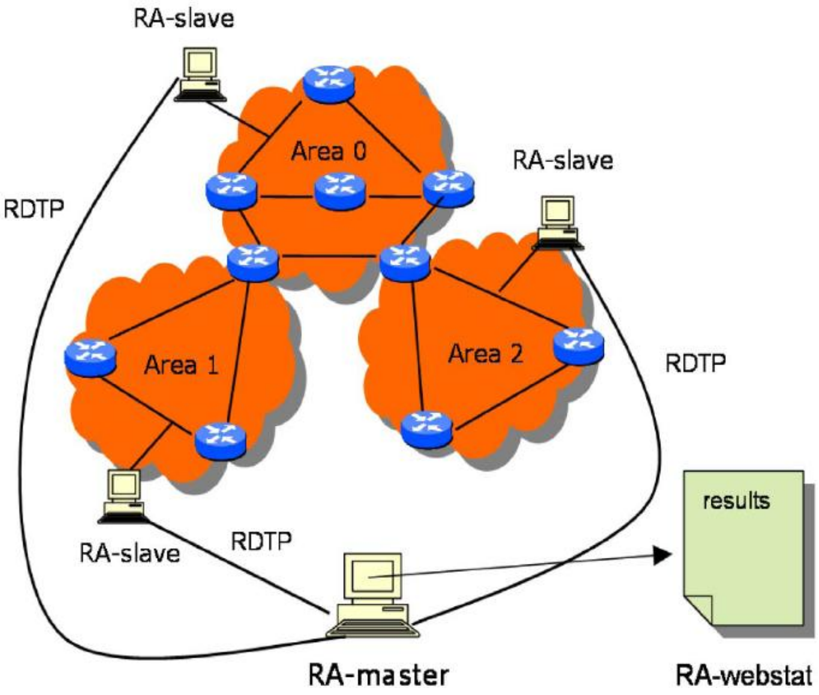


Figure 6.9: **Rtanaly architecture [1]** - Several clients are spread around the network, and a centralized host is used as a master host to gather link state logs, and to visualize the results. This network monitoring tool does not consider any topology information.

the state of an individual node or link. It is important to trace the source of the failures as shown in previous research, and described in section 3. An unstable device can lead to the generation of an excessive number of LSPs [15]. Clearly, the area where more LSP are exchanged, or LSP storm takes place, that area is suffering from instability.

One should note that the following figures account for all the reported link and gateway router failures. The data could also include reports of downtimes of links and routers which do not belong to the backbone area. As observed during 2005, the failure observations are shown in Figures 6.11 and 6.12.

The duration of downtime as shown in Figure 6.10 shows that during year 2005 the amount of links the failure of which lasted less than one hour is around 94%. According to [6,7] findings mentioned in Section 3 this indicates that if the duration of a failure be less than one hour then these could be router or protocol problems. The rest is due to optical fiber, or other communication medium problems. This can be proven by observing the router downtime durations. Link failure is characterized by failure of interfaces in the routers. One could deduce that as around 81% of times routers failed, that lasted for 1 hour or less. So by inspection one could conclude that the rest (94-81%) 13% of the link failures that lasted for less than hour could originate from some other source than physical damage in optical fibres, router or routing protocol problems.

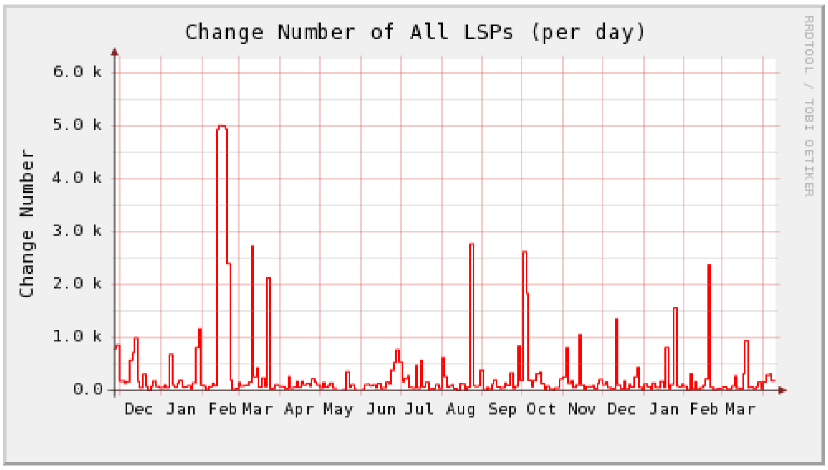


Figure 6.10: **Link-state Packet(LSP) exchange statistics in UNINETT backbone** - Rtanaly, network monitoring tool, shows the frequency of LSP exchanges, where high frequency signals problems in the state of the backbone links.

Monitoring failures using DPM

Using Defects-per-million (DPM) calculation, one can observe the frequency of failure occurrence in a network. In the following calculation, the reported node failures during year 2005 were observed. During that period 163 UNINETT nodes suffered failures. The analyzed routers are not only from the IP backbone.

- Hours per year = 8766 (Accounts for leap years)
- Number of Devices = 163
- Accumulated Hours per Year = 1,428,858 hours
- Accumulated Hours per Month = 1,428,858/12 = 119,071.5 hours
- 1,000,000 / 119,071.5 = 8.4
- 163 * 8.4 = 10911.6 Defects per Million (DPM)

6.3.2 Discussion

Observing only the rtanaly reported LSP change statistics available, one can notice that LSP has been stable for most of the time, with exception to some cases. The high peak in February 2005 is due to LSP storm, caused by some problematic flapping line between two routers which lasted for a long time. The number of LSP exchanges is dependent on the condition of links and nodes. When there are more failures in the links and nodes, the exchange of LSP is higher. But even if each transaction is analysed, this tool does not give any information how the instability could be spread. It mainly shows what links are affected by a failure of one specific link. The statistics

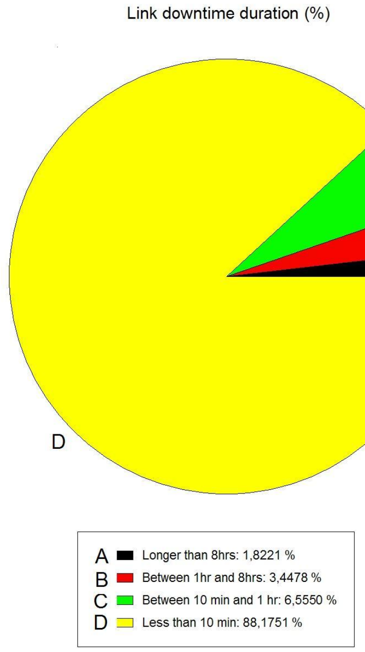


Figure 6.11: **Link statistics** - Distribution of link failure durations from 1.Jan.2005 to 31.Dec.2005, in UNINETT network topology, including all links, not only backbone links.

and reporting obtained by rtanaly does not consider overall network topology characteristics, and the importance of nodes relative to the others, but considers all nodes of equal importance. Furthermore, the number of exchanged LSPs caused by maintenance is hard to trace, as in UNINETT they have no regular maintenance period scheduled.

Monitoring failures of individual nodes and links in the network is a difficult job. Using only this information about network devices is not efficient, due to the long time needed to analyze today's complex and large networks. The overall statistics on failures over a time span can give wrong impressions and cannot be an indication of instability of the overall network. A single network device may be faulty, and may report long and frequent downtimes, and hence increase the overall system downtime value reported. A single faulty device in the network does not mean all the network is instable. Having information on how nodes are connected to each other would instead give more useful information in this aspect.

Analyzing the number of failures of devices during a period of time, as is the example of using DPM, is also misleading. If, say, the number of defects per million increases significantly, that might be because of a single or few devices, and not for the overall network. Hence, this method is hardly an indication of overall network instability. Furthermore, these calculations completely ignore the network topology characteristics which are effective indicators of stress in the network.

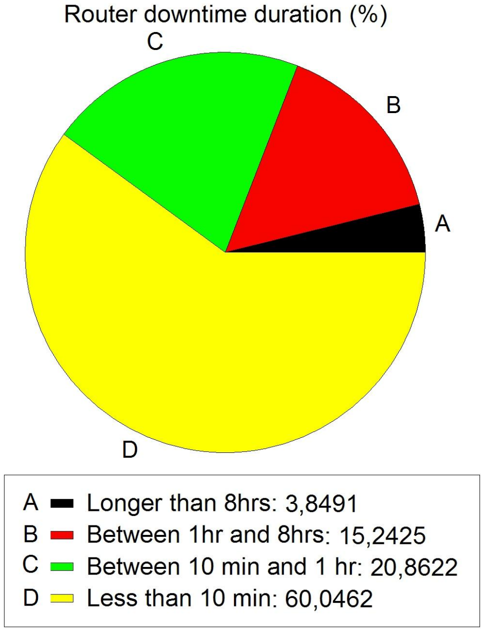


Figure 6.12: **Router statistics** - Distribution of router failure durations from 1.Jan.2005 to 31.Dec.2005, in UNINETT network topology, including all routers, not only backbone routers.

6.3.3 Monitoring stability using the proposed model

Using the same procedure as followed in Section 6.2, now the logs gathered during the last February and March of 2006 will be used for analysis. This way the network is monitored for potential instabilities taking place during this period. The network topology as shown in Figure 6.8 is used to calculate the ranking of nodes. To illustrate how the new method differs from the old method, first ranking is calculated using no other parametres but the adjacency matrix, reflecting which nodes are connected to what other nodes. The same with network connectivity calculation. In the second calculation metrics are observed and added to the equation. Only in the third proposed method is the link downtime duration considered. This is a dynamic method, in which changes in the network can be observed. Lastly, the overall stability is observed.

Data

A matrix which pictures the adjacency of nodes is created, following the topology structure of UNINETT. This is used for some of the calculations. Some calculations use a modified matrix, which includes adjacency of nodes, metrics as shown in the links of the topology, and downtime duration in links. Link downtime % is retrieved from the network maintenance website [6].

Network connectivity, χ, is calculated using the formula explained in Section 4.1.4. Uninett backbone contains 46 nodes, so N=46. The original connectivity formula

[6] http://drift.uninett.no/downs/ last time accessed in May 20, 2006

yields the value for χ_1 = 0.060376. The second method, using the link metrics as shown in the topology Figure 6.13 yields the value for χ_2 = 0.019786. This value of connectivity should be considered as the maximum possible value of connectivity, as it considers 100% availability of links. The metrics used for the second calculation are static values, and assigned by network architects, reflecting capacity of links. SLAs for ISPs claim to guarantee less than 100% availability in services. Considering that ISP SLAs try to achieve 99.999% port availability, the value above for χ_{SLA} changes to 0.019670. Observing the data, connectivity does change a little, and is a helpful indication of the narrow range between the maximum possible and SLA guaranteed provision level. The last measurement, which considers link downtimes during March 2006 as parametres in the calculation, reveals connectivity value to be χ_{03-06}=0.019615.

Network node ranking is also calculated using different values in the matrix, as in the connectivity calculation. Using the conventional method which considers a pure adjacency matrix, nodes were ranked as shown in the table in Figure 6.13. Using the second method ranking changes, as this time metrics are used in the calculation. The order by importance is shown in Figure 6.14. Second ranking could be considered as the ranking according to maximum possible provision level. This is because it considers a 100% link availability. Third method is more detailed, and measures ranking of nodes during February and March 2006.

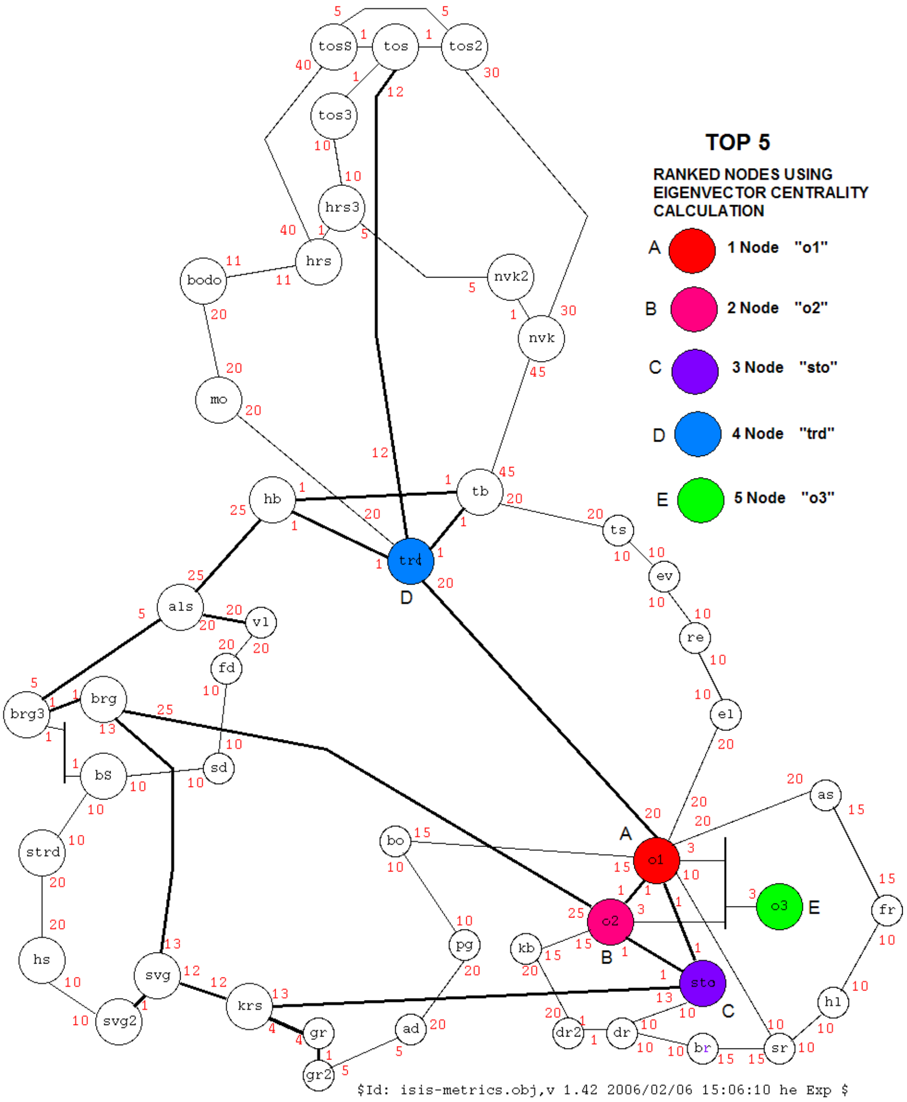
The tables in the Appendix B.1 show the cases when all links have a similar and simultaneous link failure rate. There is a low probability that all links fail simultaneously or have the same rate of link downtime. Hence, this example is only for illustration of yet a possible outcome. Observing real failures of links, data of which is registered during March and April is a better stability analysis, as it deals with real data.

Ranking results using the conventional method

The connectivity value 1 means that all nodes are connected to each other, and value 0 means all nodes are disconnected. This data obtained above show that all nodes are not connected to each other. This indicates that there is hierarchy in the way nodes are connected. This is good, especially in cases when an instability is spread in the network, typically a virus. The first method assumes any link's availability is 100% which is unrealistic. In comparison to other methods, this first estimation gives a biased value.

Nodes are ranked using the adjacency of nodes, as the only parameter. Nodes in the network can be ranked using the method described in section 4.1.2 and the script explained in appendix A.1 and the following ranking is obtained, as shown in the table below the Figure 6.13.

The main flaw of this method is that it considers links availability to be 100%. This cannot give a realistic view of what nodes are indeed more important in the network. Furthermore it does not consider any metrics, link capacity or downtime as parametres that affect importance of nodes. As such, this method is not useful to trace stability.



node rank	rank value	node name	node rank	rank value	node name	node rank	rank value	node name
1	0.50298	o1	21	0.07401	brg3	41	0.01773	sr
2	0.39298	o2	22	0.07395	dr2	42	0.01767	gr2
3	0.32889	sto	23	0.07347	als	43	0.01148	hs
4	0.31220	trd	24	0.05950	ts	44	0.01103	strd
5	0.25528	o3	25	0.05650	fr	45	0.01054	sd
6	0.17633	tb	26	0.05552	tos3	46	0.00976	fd
7	0.16013	hb	27	0.05451	re			
8	0.15941	as	28	0.05046	pg			
9	0.15905	brg	29	0.04732	hrs			
10	0.15884	e1	30	0.04260	gr			
11	0.15769	bo	31	0.04226	bodo			
12	0.15529	tos	32	0.04109	br			
13	0.13304	kb	33	0.03958	hrs3			
14	0.13184	krs	34	0.03607	nvk2			
15	0.12649	dr	35	0.03248	ev			
16	0.10099	mo	36	0.02926	svg2			
17	0.09304	tos2	37	0.02723	bS			
18	0.09122	svg	38	0.02371	v1			
19	0.08703	nvk	39	0.02115	h1			
20	0.08424	tosS	40	0.01941	ad			

🟦 top 5

Figure 6.13: **Ranking** - Ranked UNINETT nodes by importance, ignoring information on links

Ranking results using a modified method

The values of connectivity in the first and second measurement differ significantly. Clearly, not considering the capacity of links, or metrics, gives a biased value of connectivity. The speed (or ability) of one message can get to another end depends on the capacity and other characteristics of links. But this method still is not precise enough as link unavailability parameter is ignored.

In the second improved method nodes are ranked using the adjacency of nodes and statically assigned link metrics as parametres. The second method still assumes that the availability of links is 100%. The second improved ranking method shows different results, as shown in 6.14.

As can be noticed from the Figure 6.14, and the table shown in it, the second method shows a better network hierarchy. It identifies nodes which are closer to the most important nodes. Nodes which provide link redundancy to most important nodes appear to be ranked higher in the hierarchy. Thus, the method evaluates better the hierarchy of nodes in a network, but does not develop in time as link metrics are assigned manually. In order to observe stability of the network, a dynamical parametre is needed. This parameter should change in time.

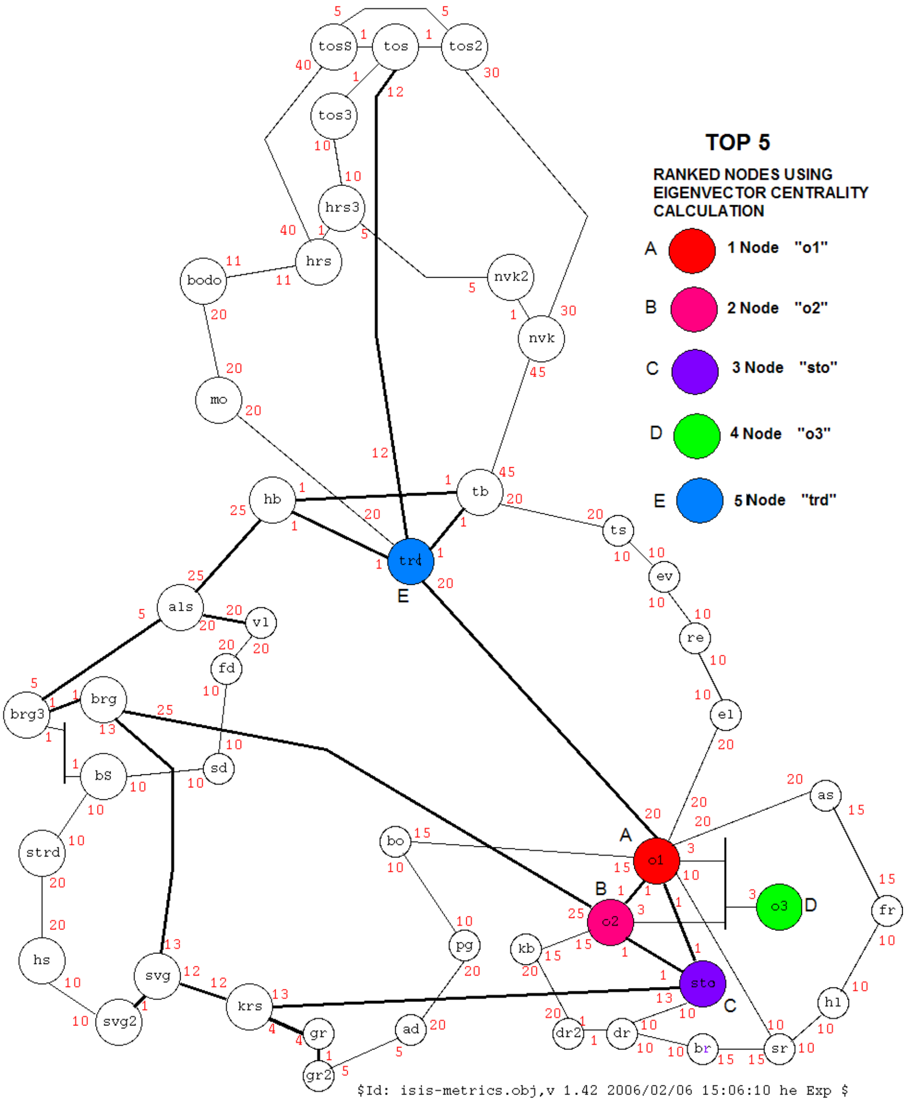
Ranking results using the proposed method

The proposed method does not include only the adjacency of nodes, or statically assigned link metrics, but also a parameter that changes in time: downtime percentage of links. Assuming that any link's availability is 100% and there is no packet loss is not realistic, and does not give reflect the real picture. Including packet loss in the equation would increase the precision of the calculations, but it is not available for analysis in this case. A random 1% packet loss is assumed instead.

The values retrieved in the second network connectivity measurement, χ_2, and third case, χ_{SLA} , are not as different as in the first case. Instead, third proposed method is more precise, and considers link downtime as a dynamical parametre. The value of χ_{SLA} changes a little, when downtimes during March 2006 were added to the measurement. The obtained value χ_{Feb-06} = = 0.019639 shows that even if the difference is small, it still confirms that the hypothesis is correct. This proves that link downtime indeed affects the connectivity of the network. The higher the downtime, the lower the connectivity.

Link downtime percentage of links for February and March 2006 added to the equation change the network ranking values. First, compared to the SLA-compliant ranking, the ranking during February 2006 differs as shown in Table B.1. Second, link downtimes during March 2006 change the ranking to the form as shown in the same Table B.1. As the rtanaly LSA packet rate shown in Figure 6.10 it shows no signs of large size of LSA storms taking place. This is also confirmed by the obtained graph, using the proposed stability monitoring method, where no signs of instability are noted (see Figure 6.15).

If the Figure 6.15 is enlarged, to monitor deeper for any signs of instabilities, we notice that nodes ”als”, ”brg3” have suffered from instabilities during the two months



Rank	Rank Value	Node	Rank	Rank Value	Node	Rank	Rank Value	Node
1	0.56192457459	o1	21	0.00942167833224	tos3	41	0.000169212887428	ad
2	0.559643659814	o2	22	0.00469828410346	bS	42	0.000166248783021	ev
3	0.539942761983	sto	23	0.00318754617463	gr	43	1.22795271283e-05	fd
4	0.17940646921	o3	24	0.00303321331527	mo	44	4.70705666168e-05	hs
5	0.126320271493	trd	25	0.00280786812835	ts	45	5.96716258121e-05	v1
6	0.116690993394	tb	26	0.00247465420501	als	46	9.49284927416e-05	bodo
7	0.116664016774	hb	27	0.0021289787907	br			
8	0.0270973514189	sr	28	0.00188030689168	nvk			
9	0.0262998125805	dr	29	0.00180320775314	svg			
10	0.0203859313724	krs	30	0.0015458836054	gr2			
11	0.0195543947828	tos	31	0.0013241237039	h1			
12	0.018217473163	kb	32	0.000978111129706	nvk2			
13	0.0180188221576	bo	33	0.000868749338556	pg			
14	0.0154856229392	brg	34	0.000867584163073	svg2			
15	0.0135143538845	e1	35	0.000789647402054	hrs3			
16	0.0134986980553	as	36	0.000656506277359	re			
17	0.0130578942763	dr2	37	0.000507716271469	hrs			
18	0.0104109668445	tos2	38	0.000495393540861	fr			
19	0.0103890809934	tosS	39	0.000226591192165	strd			
20	0.00974525267646	brg3	40	0.000226051049689	sd			

Top 5 🟦

Figure 6.14: **Ranking** - Ranking of UNINETT nodes by importance, using metrics information on links

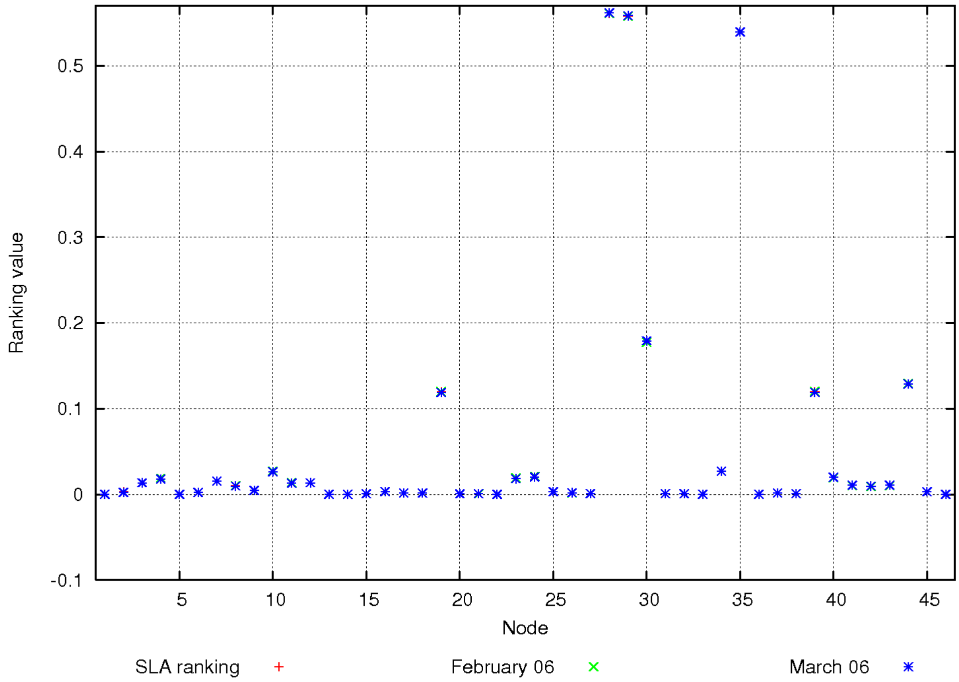


Figure 6.15: **Live-network analysis** - This graph shows the analysis of stability in UNINETT network, by analysing failures in links during February and March of 2006. The redundancy of links and the small amount of failures, mainly in regional nodes shows that there are no signs of instability in that period.

of observation. Nodes ″bo″ and ″dr2″ on the other hand suffered from link instabilities in February only, but no such signs were noted in March.

Similar link failure rates

There is a low probability that links fail simultaneously, and have a similar failure rate. As seen in the appendix B.1 the tables show an interesting phenomenon when link failure percentages are similar. Due to increased link failure rate some nodes which are ranked high in the network hierarchy become less important, i.e. less loaded, (see Figure 6.17)and the other way round (see Figure 6.18). This is explained with the fact that more important nodes (and links) suffering link failures cannot forward traffic due to increased failure rate in their surrounding links, and instead less important nodes have to find alternative links to do that.

The hypothesis posed previously proved to be partly correct, but a slightly different result is obtained. In Figure 6.18 a value of a high-ranked node is increased. This means that that node cannot share its load with the rest high-ranked nodes which suffered link failures, and as a result its load is increased.

After observing the overall connectivity value, the hypothesis that the connectivity value will decrease with the link uptime value decreasing, proved to be correct.(see

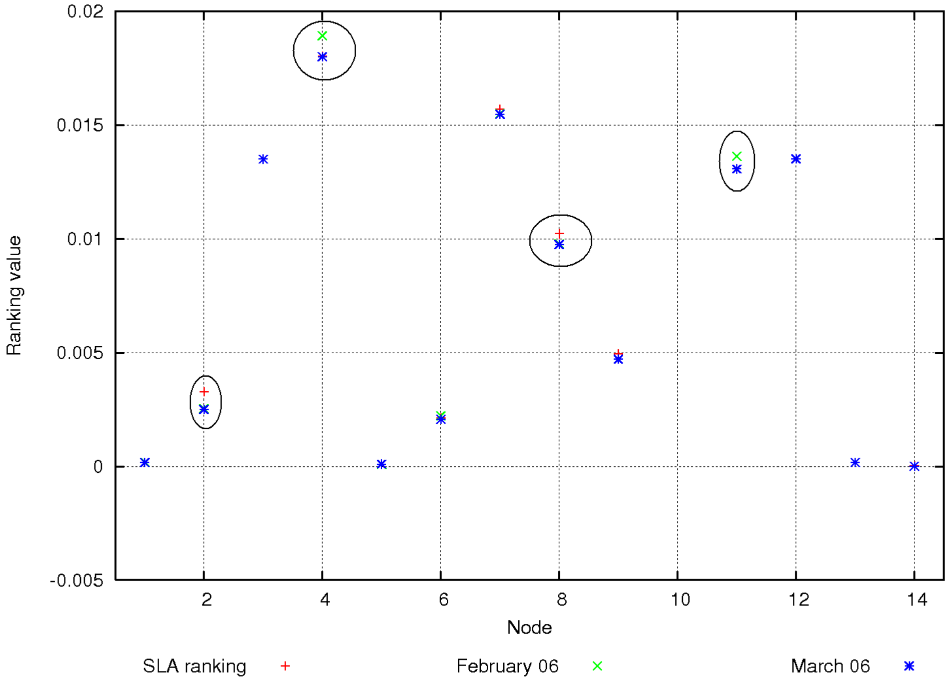


Figure 6.16: **Live-network analysis zooming** - If Figure 6.15 is zoomed enough, we can notice minor instabilities in the circled nodes corresponding to ″als″, ″brg3″, ″bo″ and ″dr2″.

Figure 6.19) This is logical, as connectivity is strongly dependent on the ability of a message to pass through a link. As the probability is low for this scenario to occur, this observation is valid only for illustration, and considering worst case scenario.

6.4 Tips on improving the overall network stability

Stability of a network is affected by the frequency of device failures. Failures can be prevented if careful attention is paid to several aspects, among many: Uninterruptible Power Supply (UPS), highly restricted human access to main devices, routing configuration tuning, traffic engineering (TE) using Multi-protocol Label Switching(MPLS), etc.

6.4.1 UPS

UPS, in cases when there are problems in power supply, the lack of UPS will cause the network services to suffer significant instability rate. Even though this is one of the requirements that typical ISPs carefully address, it is still worth mentioning as a crucial stability factor. Providing redundant power supply sources is extremely important.

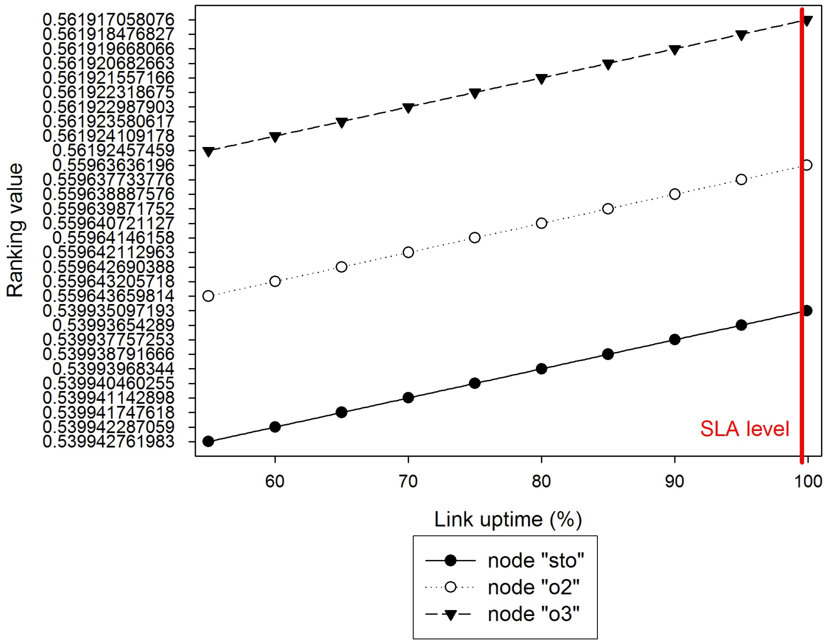


Figure 6.17: **Simultaneous failures** - This graph shows the rare case when all network links fail simultaneously and with a similar downtime percentage. The ranking value of the three top-most nodes in the network change in the same way.

6.4.2 Human access

Human access issues often identified as the main source of errors. This observation is logical. Unauthorized physical access to main devices should be banned. Furthermore, links which are used for distribution between central and regional nodes should be protected and labelled to prevent accidental or physical damage.

6.4.3 Routing configuration tuning

As of now, the IS-IS routing configuration, responsible for exchanging routing information in UNINETT network, is only configured using the default parametres. Parametre tuning is scheduled to be carried out in the near future.

The speed of convergence is important to the stability of networks. Should the convergence be fast, it means that the instability will be localized, and fast recovery will take place. One should bear in mind that network protocol tuning is subject to the topology where the protocol tuning is applied, thus it may not show the expected results in any network. The convergence time varies, and sometimes can be up to a couple of minutes. There are many parametres which if tuned could speed up the routing convergence period to sub-seconds instead of minutes. In the case of Francois et al [15] observations, the tuning performed very well, and sub-second convergence was achieved. Some of the parametres that could be changed would be:

- *Hello interval*, previous research has shown that decreasing the value of this parametre helps improve convergence,

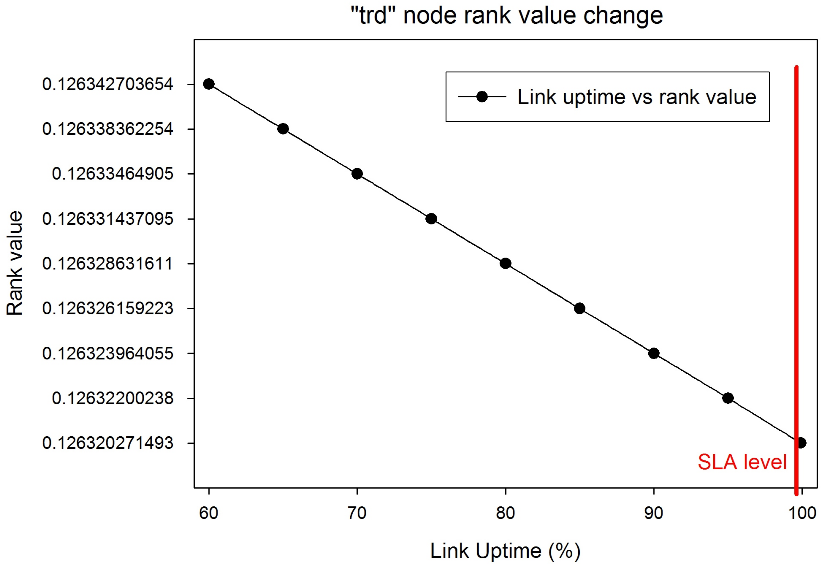


Figure 6.18: **Simultaneous failures** - This graph shows the rare case when all network links fail simultaneously and with a similar downtime percentage. In this case, the ranking value of important node such as node ”trd” increases with the rest of links failing around the network. This is explained with the fact that this node covers a vast amount of nodes around it, and is not connected to high-capacity and high-ranked nodes such as is node ”o1”

- *SPF delay* decreased value helps in achieving fast convergence times.
- *Incremental SPF*, this IS-IS parametre could be used, so instead of calculating the SPF tree for the overall network topology, only portions of the network which change get notified of the link state [43].
- *Fast flooding* In order not to wait for the SPF delay timer to expire, this command is used to start the SPF calculation immediately as the link failure is sensed in the network [44].

One should observe the CPU load after such tuning is done. Previous research has shown that too much tuning could decrease the stability threshold. This would be the case when the LSA storm size is large. Decreasing the abovementioned timers too much will significantly increase the load and calculations on the router CPUs. Depending on the capacity of the routers, some routers might be disabled completely due to overload.

6.4.4 Traffic Engineering

Dynamically changing metrics in link-state protocols could be harmful to the stability of the networks. Instead, new methods could be applied for fast switching and traffic engineering (TE) [13, 34] such as MPLS to additionally protect networks. MPLS supports the rerouting of traffic around a failed link or router quickly enough to not affect the users of the network [30]. TE can be performed using the optional parametres that

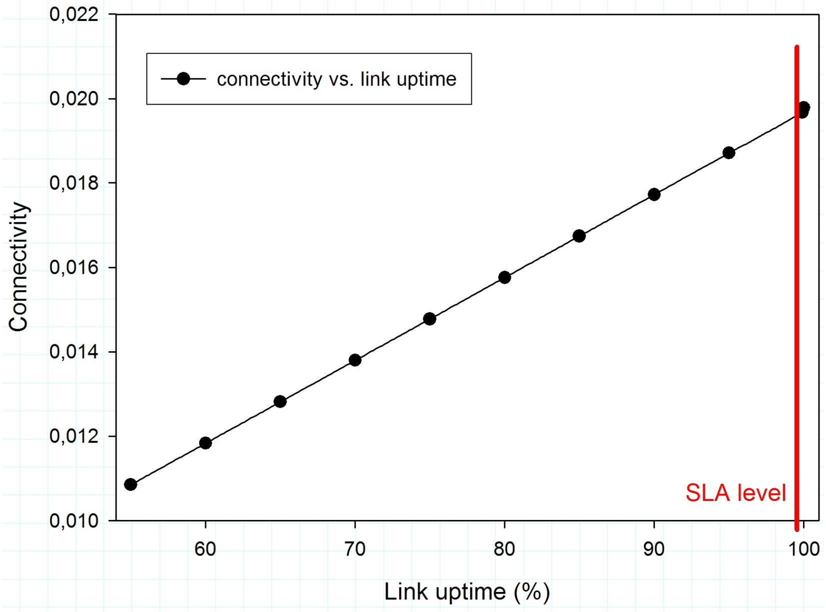


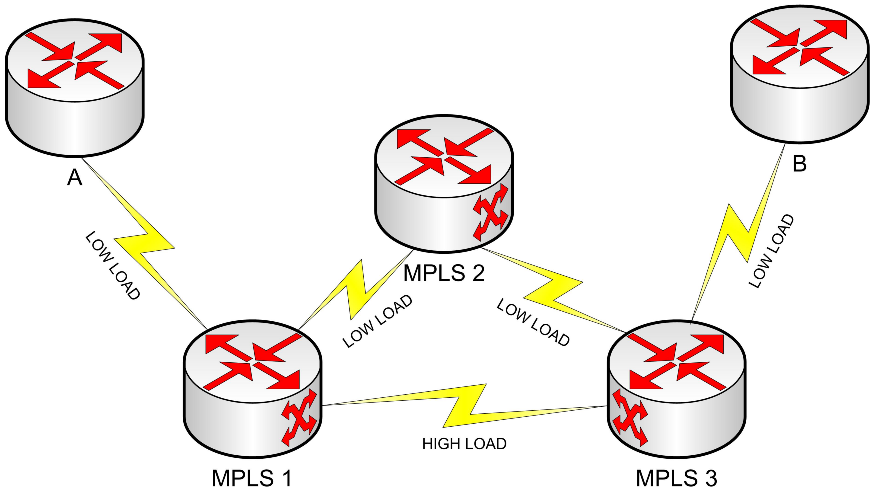
Figure 6.19: **Connectivity** - As expected connectivity value, χ, is proportional to the link failure rate, and as such changes linearly.

are available in LSP optional fields. MPLS traffic engineered packets make use of simple labels, which are used to switch packets according to those labels, thus reducing lookup overhead. Additionally, actual link loads of the traffic are measured, to adjust the routing of the traffic to fit the actual bandwidth available [45]. Some significant advantages of using MPLS in IP backbone networks are [30,45]:

- traffic engineering is integrated in layer 3, thus optimizing IP traffic routing, given the backbone constraints,
- makes best use of links, by fitting IP traffic to the available link bandwidth, this way load balancing is achieved, and packet loss could be prevented, (see Figure 6.20)
- can classify traffic by select specific routes for certain traffic. Should there be a need for that, specific traffic can be prioritized, for e.g. routing update packets are prioritized over normal traffic packets.
- shifts the traffic load from overutilized parts to underutilized parts of the network, directed so by traffic destination, traffic type, traffic load, etc.

After a survey was conducted in year 2005, to see reasons why companies choose MPLS, the vast majority of companies were either making their initial deployment of VoIP or expanding their VoIP deployment are also planning on expanding their Quality of Service(QoS) policies. [7] Though this is the case, MPLS could be applied

[7] http://www.networkworld.com/newsletters/frame/2005/0418wan1.html last time accessed on May 20, 2006



A – MPLS 1 – MPLS 3 – B *Normal path*

A – MPLS 1 – MPLS 2 - MPLS 3 – B *MPLS Traffic engineered path*

Figure 6.20: **MPLS traffic engineering** - An illustration of the ability of MPLS traffic engineered router to redirect packets around a less loaded link, instead of risking to lose any packets by transmitting over an overloaded link

to networks which want improved traffic engineering, and fast recovery from device failures [46].

Another way of utilizing TE capabilities is using *Netscope*. According to Feldmann et al [14] *Netscope*, a unified set of software tools for managing the performance of IP backbone networks, i.e. traffic engineering, has that capability.

6.5 Future work

The precision of the method of observing ranking values and connectivity of a network nodes can be improved should more parametres be included in the calculation. In order for the parametres to be included, data logs should be available. Data logs should be centralized in a way to feed the script in the most precise way. This section suggests several tips on how this method best can be utilized.

- **Downtime percentage**, the data on downtime percentage of links should exist separately for backbone links,

- **Packet loss**, the data on packet loss on backbone links should exist separately for backbone links,

- **Delay**, data logs on packet delay in the IP backbone should be available for stability measurement, is a key metric in data network performance and an important parameter in Internet service providers (ISPs) service level agreements [47].

- **Updates**, should updates occurr in the backbone links or nodes, that should reflect in the metrics, and data logs.
- **Maintenance**, the period when maintenance is carried on should be indicated in the data logs. This way the downtime caused by regular maintenance is ignored when measuring duration of failures.
- **DPM**, information on the rate of change of Defects per Million could also be an indicator of change in network stability.

The time for completion of this thesis was short, and did not leave enough time for developing a web-based tool to extract the information. This tool would automatically collect link-state information, and update the calculations in a continuous fashion, to reveal signs of instabilities almost in real-time. The script would be web-based, i.e. accessible from any site in the network.This could be used to visualize the changes in ranking. The rate of change could be indicated with different colours, to distinguish the regions of significant instability from those which are within the tolerable threshold, within the SLA provision level.

The concepts introduced in this thesis will be expanded in the future to include network flow information and ways how it affects the stability. In this thesis only packet loss was introduced as a way to characterize the effects on the stability caused by the disturbance in the packet flow. Yet that data was not available as of now, to be incorporated in the effective expected adjacency equation 4.5.

Future work will also include analysis of more conventional network monitoring tools and ways how this concept could be incorporated in them. Additionally, in the near future we will observe network tuning, to see the effects in the live network. UNINETT has planned to carry out network tuning in the near future.

Chapter 7

Conclusions and Discussion

"Stability is (indeed) unstable." - Hyman Minsky, an American theoretical economist

When network routes are established, and routing of information is successfully exchanged between nodes in a network, measures should be taken to keep the system running in that state. This state represents the state which by networking policy is defined as the wanted or the stable state [2]. As the research results have shown, the stable state cannot last for long periods of time, as a result of failures occuring, i.e. environment tends to change the stable state. In response, proactive and reactive measures are taken by network operators to make the system converge back to the wanted state.

A network is stable if its service provisioning is available close to 100% of the time end-users need to access those services. The users expect the level of services provisioned to them to be at least the minimum of the SLA level. To approach this high demand fast recovery of services should be prioritized. If service downtime occurs, that is usually manifested with packet loss. Packet loss means money loss. In this context "money" loss does not necessarily mean only loss in cash value. "Money" could be users' credibility, damage to users' assets, time spent to recover from the failure, etc [2]. Therefore, stability of services is very important for complex network systems. This thesis is an additional attempt to increase the awareness on the stability issues related to computer networks by introducing a new method to trace the stability of networks.

Stability of the networks can be studied by monitoring the instability caused in networked nodes should any link or node of the network experience problems. Observations of such changes are done using network monitoring tools. Conventional tools used in network monitoring almost ignore characteristics of the network topologies. The model explained in this thesis can be incorporated in a network monitoring tool, to get a better view on rate of change in the stability of networks. A model for finding most important nodes in the network is proposed. In that way nodes are ranked, taking into consideration more information than static data such as the adjacency of nodes in the network. Additionally the duration of failures was used as an indication of the origin of errors. This method can be suitable in monitoring stability in networks of the size of our case study, UNINETT network. It is difficult to apply it in a larger scale network, say measuring the stability of Internet infrastructure, as there is no clear hierarchy in its topology, and its links are very dynamic.

Several tips on how to speed up the network routing convergence have been proposed. Due to the lack of time, these proposals could not be tested in a live-running UNINETT network.

The previous research done in the field of IP backbone failures, as mentioned in Chapter 3, has shown that link failure durations of less than one hour indicate routing protocol problems. In the observed network data logs provided by UNINETT for year 2005, these failures accounted for 94% of the reported failures. Taken with a level of uncertainty, the method's analysis and results introduced in this thesis work identify that around 94% of the factors that change stability of the overall network, as caused by the routing protocol problems. It also shows how nodes are affected by changes occurring in links. Some nodes become more loaded, their ranking value increases, and some nodes become less loaded, their ranking value decreases. Significant decrease in load of nodes means nodes are not utilized well. Significant increase in load of nodes means that there is risk that those nodes could exceed their capacity level, and may not be able to provision the expected services in the near future. In the latter case, packet loss is expected to take place.

The proposed stability monitoring method is unique, and to the best of our knowledge, has not been used before. The emphasis on this thesis is that the characteristics of the topology of the network should be considered, to have an effective view on what areas can be a source of instability. Conventional network monitoring tools, for e.g. *Nagios* [10], *Munin* [11], *Rtanaly* [1] etc, measure both the reliability of the nodes (routers) and the links (lines of transmission) in a network, but these data are never (to our knowledge) put together with a model of the actual topology in order to predict most likely causes of failures and how they affect nodes surrounding them.

The calculations derived from the proposed model give another information which was not the initial intention of thesis. The ranking of nodes shows also how vulnerabile nodes are. The network should have protection-levels according to the rank values, the higher the rank, the more vulnerable and the more security measures should be applied for that node. Similar studies were analysed by Stang et al [48] but they disregard the link-states in the detail our method does. IP backbone operators could adapt this method to observe the stability in their networks.

The precision of the ranking method has a level of uncertainty, as there are many other parametres which can be considered when analyzing the stability of the network. Packet loss is only one of the parametres left out from this analysis. Observing the ranking of nodes in a network, and the dynamics that ranking changes indeed gives insights on the level of overall network instability. The improvement of the method is planned in the future work.

Bibliography

[1] S. Zhang and K. Kobayashi. Rtanaly:a system to detect and measure igp routing changes. *Operations and Management in IP-Based Networks: 5th IEEE International Workshop on IP Operations and Management, IPOM 2005, Barcelona, Spain*, 3751, October 2005.

[2] M. Burgess. *Analytical Network and System Administration*. Wiley, 2004. ISBN 0-470-86100-2.

[3] S. Nelakuditi, S. Lee, Y. Yu, and Z. Zhang. Failure insensitive routing for ensuring service availability. *Proceedings of International Workshop on Quality of Service (QoS)*, 2003.

[4] M. Burgess. System administration research 1. *;login: The Magazine of USENIX and SAGE*, June 2000.

[5] CISCO. *IP Routing Fundamentals*. http://www.cisco.com.

[6] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of link failures in an IP backbone. *ACM SIGCOMM Computer Internet Measurement Workshop*, November 2002.

[7] A. Markopoulou, G. Iannaccone, C. Chuah, S. Bhattacharyya, and C. Diot. Characterization of failures in an IP backbone. *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, 4:2307 – 2317, March 2004.

[8] E. Marilly, O. Martinot, H. Papini, and D. Goderis. Service level agreements: a main challenge for next generation networks. *2nd European Conference on Universal Multiservice Networks, IEEE CNF*, pages 297–304, 2002.

[9] Fanglu Guo, Jiawu Chen, Wei Li, and Tzi cker Chiueh. Experiences in building a multihoming load balancing system. *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, 2:1241 – 1251, 2004.

[10] R. C. Harlan. Network management with Nagios. *Linux Journal*, 111, 2003.

[11] J. B. Carter, J. K. Bennett, and W. Zwaenepoel. Implementation and performance of Munin. *ACM SIGOPS Operating Systems Review*, 25 – 5:152 – 164, 1991.

[12] M. Burgess, G. Canright, and K. Engø. A graph theoretical model of computer security: from file access to social engineering. *International Journal of Information Security*, 3:70–85, 2004.

[13] Fawduche D.O. MPLS and traffic engineering in IP networks. *Communications Magazine, IEEE*, 37 – 12:42 – 47, December 1999.

[14] Feldmann A., Greenberg A., Lund C., Reingold N., and Rexford J. Netscope: traffic engineering for IP networks. *Network, IEEE*, 14 – 2:11 – 19, April 2000.

[15] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure. Achieving sub-second IGP convergence in large IP networks. *ACM SIGCOMM Computer Communication Review*, 35, July 2005.

[16] G. Canright and K. Engø-Monsen. A natural definition of clusters and roles in undirected graphs. *Science of Computer Programming*, 53:195, 2004.

[17] M. Burgess and G. Canright. Scaling behaviour of peer configuration in logically ad hoc networks. *IEEE eTransactions on Network and Service Management*, 1:1, 2004.

[18] R. W. Callon. Use of OSI IS-IS for routing in TCP/IP and dual environments, rfc 1195. December 1990.

[19] R. Perlman. A comparison between two routing protocols: OSPF and IS-IS. *IEEE Network*, 5:18–24, September 1991.

[20] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot. IGP link weight assignment for transient link failures. *In International Teletraffic Congress*, August 2003.

[21] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20 4:756–767, 2002.

[22] B. Fortz and M. Thorup. Robust optimization of OSPF/IS-IS weights. *Proceedings of the International Network Optimization Conference*, pages 225–230, 2003.

[23] P.S.Weygant. *Clusters for High Availability: A Primer of HP Solutions, Second Edition*. Prentice Hall, 2001. ISBN 0-13-089355-2.

[24] C. Oggerino. *High Availability Network Fundamentals*. CISCO 2001.

[25] F. Giroire, A. Nucci, N. Taft, and C. Diot. Increasing the robustness of IP backbones in the absence of optical level protection. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Conference Proceedings*, 1:1 – 11, March, April 2003.

[26] C. Diot, G. Iannaccone, A. Markopoulou C. Chuah, and S. Bhattacharyya. Service availability in IP networks. *Sprint ATL Research Report Nr. RR03-ATL-071888*, July 2003.

[27] A.S. Tanenbaum. *Computer Networks, Third Edition*. Prentice Hall, 1996. ISBN 0-13-394248-1.

[28] C. Paquet and D. Teare. *Building Scalable Cisco Networks*. Ciscopress, 2000. ISBN 1578702283.

[29] D. Oran. OSI ISIS Intradomain Routing Protocol RFC 1142. *Digital Equipment Corp.*, December 1990.

[30] V. Alwayn. *Advanced MPLS Design and Implementation*. Cisco Press, 2001.

[31] Abe Martey and Scott Sturgess. IS-IS network design solutions. *Cisco Press*, 2002.

[32] G. L. Choudhury, A. S. Maunder, and V. D. Sapozhnikova. Faster link-state IGP convergence and improved network scalability and stability. *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks, 2001. LCN 2001*, 2001.

[33] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. *Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2000*, 2:519 – 528, March 2000.

[34] B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communications Magazine*, pages 118–124, October 2002.

[35] Miyamura T., Kurimoto T., and M. Aoki. Enhancing the network scalability of link-state routing protocols by reducing their flooding overhead. *Workshop on High Performance Switching and Routing, 2003, HPSR. IEEE CNF*, pages 263 – 268, June 2003.

[36] Jie Wu, Xiaola Lin, Jiannong Cao, and Weijia Jia. An extended fault-tolerant link-state routing protocol in the internet. *Proceedings from Eighth International Conference on Parallel and Distributed Systems, 2001. ICPADS 2001. IEEE CNF*, pages 331 – 337, June 2001.

[37] Anindya Basu and Jon G. Riecke. Stability issues in OSPF routing. *ACM SIGCOMM Computer Communication Review*, 31 – 4, October 2001.

[38] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot. An approach to alleviate link overload as observed on an IP backbone. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Conference Proceedings*, 1:406 – 416, March – April 2003.

[39] R. Kone and H. Zhou. End-to-end availability analysis of physical network. *The Fourth International Conference on Computer and Information Technology, 2004. CIT '04. IEEE Conference Proceeding*, pages 668 – 673, September 2004.

[40] Choudhury T. and Pentland A. Sensing and modeling human networks using the sociometer. *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings. IEEE CNF*, pages 216 – 222, October 2005.

[41] E. Costenbader and T. W. Valente. The stability of centrality measures when networks are sampled. *Elsevier B.V., Social Networks, Science Direct*, 25:283 – 307, 2003.

[42] T.H. Haveliwala. Topic-sensitive PageRank: a context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15 – 4:784 – 796, August 2003.

[43] CISCO. *Cisco IOS IP Routing Protocols Configuration Guide, IS-IS Incremental SPF*. http://www.cisco.com.

[44] CISCO. *Cisco IOS IP Routing Protocols Configuration Guide, IS-IS Fast flooding*. http://www.cisco.com.

[45] CISCO. *CISCO MPLS Controller Software Configuration Guide*. Ciscopress, 2001.

[46] I. Hussain. Understanding high availability of IP and MPLS networks. January 2005.

[47] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, and C. Diot. Measurement and analysis of single-hop delay on an IP backbone network. *IEEE Journal on Selected Areas In Communications*, 21/6, August 2003.

[48] Tuva Hassel Stang, Fahimeh Pourbayat, Mark Burgess, Geoffrey Canright, smund Weltzien, and Kenth Eng. Archipelago: A network security analysis tool. *Proceedings of the 17th Large Installation Systems Administration(LISA) Conference, San Diego, CA, USA*, 2003.

Chapter 8

List of Abbreviations

CPU	Central Processing Unit
DPM	Defects per Million
FTP	File Transfer Protocol
IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IP	Internet protocol
IS-IS	Intermediate-System to Intermediate-System routing protocol
ISH	Intermediate-System Hello
ISO	International Standards Organization
ISP	Internet Service Provider
LSA	Link State Advertisement
LSP	Link State Packet
MPLS	Multi-protocol Label Switching
MTBF	Mean Time Between Failures
MTTR	Mean Time to Repair
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PDU	Protocol Data Units
RFC	Request for Comments
SAP	Service Access Point
SLA	Service Level Agreement
SPF	Shortest Path First
Telcos	Telecommunications Company
TCP	Transmission Control Protocol
TE	Traffic Engineering
UDP	User Datagram Protocols
UPS	Uninterruptible power supply

Appendix A

Calculating Eigenvector Centrality script

A.1 Calculating Eigenvector Centrality script

A group of scripts is created in order to automate the Eigenvector centrality. The archive containing files can be found in [1]. The script makes use of GNU Octave, version 2.1.69, which ″is a high-level language, primarily intended for numerical computations.″[2] and provides the commands needed to calculate the Eigenvector and Eigenvalue of a given matrix.

This group of scripts contains the following files:

- *Readme* which explains what the script does.
- *main.sh* is the main script that calls the octave commands, gets input redirected and formats output, which is a ranked list of nodes.
- *file.m* is the input file for Octave, it is where operations on the matrix are configured.
- *matrix.oct* contains the Adjacency matrix which will be used to rank nodes in a network.
- *matrix_evect.out* is the Eigenvector log, an output Octave produces.
- *matrix_eval.out* is the Eigenvalue log, an output Octave produces.
- *eig_col.pl* is a perl script used to inspect which column contains the highes Eigenvalue.
- *eig_col* is an output file from the perl script above which shows which Eigenvector should be used for ranking.

[1] http://student.iu.hio.no/ s126316/centrality/Cetrality.rar

[2] http://www.gnu.org/software/octave/

- *ranked_nodes1* this is the last output file, which shows the ranked nodes

The format of the and its format is:

```
Rank Value       Node    Rank
```

where,

Rank Value is the value as shown by the Eigenvector with highest Eigenvalue,
Node is the node name
Rank is the position in the ranked list

In order to run the script, the archive provided above should be extracted to a directory. It assumes that the GNU Octave software is installed, and its path is *"/usr/bin/octave"*. If not, that should be fixed in the "main.sh" file. The only input file to be customized is "matrix.oct" which should reflect the Adjacency matrix of the network. The improved version of ranking method takes other parametres as well, which should be combined to reflect the matrix elements to be analyzed.

Running the script is trivial, one should only execute the following commands, after matrix.oct is customized:

- `cd centrality`

  enter the directory where the archive is extracted.

- `chmod a+x main.sh`

  make the main file executable,

- `sh main.sh`

  make the main file executable,

- `vi ranked_nodes1`

  view the ranked nodes.

A.1.1 Main files

The main script that is executed is *main.sh*, which is a script that executes an octave input file *file.m*, and fixes the format of the output file. The *main.sh* looks as follows:

```
#!/bin/sh

#call file.m which calls octave commands.
/usr/bin/octave file.m

#two outputs are generated: matrix_eval.out and matrix_evect.out
# an awk script is called to find the column of the highest eigenvalue

#remove unnecessary text in output evect_matrix
sed ’1,5d’ matrix_evect.out > matrix_evect1.out
sed ’s/ˆ \+//’ matrix_evect1.out > matrix_evect2.out
sed ’s/,0//g’ matrix_evect2.out > matrix_evect3.out
sed ’s/(//g’ matrix_evect3.out > matrix_evect4.out
sed ’s/)//g’ matrix_evect4.out > matrix_evect5.out

#remove unnecessary text in output eval_matrix
sed ’1,5d’ matrix_eval.out  > matrix_eval1.out
sed ’s/ˆ \+//’ matrix_eval1.out > matrix_eval2.out
sed ’s/,0//g’ matrix_eval2.out > matrix_eval3.out
sed ’s/(//g’ matrix_eval3.out > matrix_eval4.out
sed ’s/)//g’ matrix_eval4.out > matrix_eval5.out

#eig_col is the column which has the highest eigenvalue
#it is used to rank the respective eigenvector column.

/usr/bin/perl eig_col.pl

#append names to nodes before they are ranked/sorted
paste matrix_evect5.out node_names > matrix_evect6.out
echo -e "Rank Value             Node    Rank" > ranked_nodes
echo -e "=======================================\n" >> ranked_nodes

#echo "Rank         Rank Value          Node name" > ranked_nodes
#echo "===============================" >> ranked_nodes
##the following script sorts/ranks the needed column in matrix_evect.
sort -nr -k $(cat eig_col) matrix_evect6.out \\
|awk ’{print $’$(cat eig_col)’, "\t\t"$47}’ >> ranked_nodes

paste ranked_nodes list1 > ranked_nodes1

exit
```

The file.m input file that feeds the Octave software is:

```
#Load the matrix values from matrix.oct to the variable matrix.
load -force matrix.oct matrix

#display matrix
matrix
save_precision = 8

#output format of the values in the matrix
format short

#calculate the eigenvector and eigenvalue of the matrix
[evect,eval]=eig(matrix)

#save the eigenvector and eigenvalue in matrix.out
save -ascii matrix_evect.out evect
save -ascii matrix_eval.out eval
```

After the eigenvectors and eigenvalues of the matrix are calculated, **eig_col** is needed to find the eigenvector which has the highest value of the eigenvalue. The following perl script scans through the matrix to find that eigenvector, and displays the number of the column that contains it:

```
#!/usr/bin/perl

$filename="matrix_eval5.out";

open (FP,$filename) || die("Error: could not open file $filename");

     @content=();
     while($file=<FP>)
    {
        @cols=split(/ /,$file);
        $size=scalar(@cols) . "\n";
        push(@content,$file);
}

close(FP);

$counter=0;
@max=();
       while($counter<$size)
        {
        $counter+=1;
        $highest=0;
          for $line (@content)
          {
$line=˜s/\n//g;
@row=split(/ /,$line);
$number=$row[$counter-1];
if($number>$highest){
$highest=$number;
          }
        }

     push(@max,$highest);
}

        $tmp1=0;
        $colnumber=0;

         for(0..$#max)
         {
          $key=$_;
          if($max[$_] > $tmp1)
          {
           $tmp1=$max[$key];
           $colnumber=$key+1;
}
         }

print ‘echo $colnumber > eig_col‘;
```

The output **eig_col** is then used by **main.sh** to rank the nodes from that column.

Appendix B

Different ranking results

B.1 Different ranking results

This section includes several results obtained in observing ranking of nodes.

Node	Δ(SLA vs 95%)	Δ(SLA vs 90%)	Δ(SLA vs 85%)	Δ(SLA vs 80%)
o1	-8.28246E-05	-0.000176887	-0.000282367	-0.000401462
o2	-8.11402E-05	-0.000173222	-0.000276399	-0.000392792
sto	-8.79582E-05	-0.000187865	-0.000299862	-0.000426291
o3	-8.81624E-05	-0.000187852	-0.000299959	-0.000426523
trd	0,0013702	0,0029232	0,0046610	0,0066182
tb	0,0014843	0,0031664	0,0050489	0,0071691
hb	0,0016202	0,0034566	0,0055116	0,0078264
sr	-8.88603E-05	-0.000189835	-0.000303063	-0.000430943
dr	-9.79026E-05	-0.000209093	-0.000333727	-0.000472877
krs	-6.55977E-05	-0.0001398	-0.000222707	-0.000314344
tos	0,0013283	0,0028351	0,0045196	0,0064165
kb	-8.64577E-05	-0.000186839	-0.000297591	-0.000422549
bo	-8.77671E-05	-0.00018974	-0.000302326	-0.000429484
brg	0,0193654	0,0415140	0,0665386	0,0950331
e1	-8.87841E-05	-0.000189664	-0.000302792	-0.000430561
as	-8.88049E-05	-0.000189712	-0.000302869	-0.000430669
dr2	-0.000103487	-0.000221141	-0.00035297	-0.000500328
tos2	0,0013220	0,0028217	0,0044982	0,0063860
tosS	0,0013217	0,0028210	0,0044971	0,0063844
brg3	0,0642182	0,1376644	0,2206453	0,3151299
tos3	0,0013223	0,0028223	0,0044992	0,0063874
bS	0,0642119	0,1376509	0,2206236	0,3150990
gr	-7.52281E-05	-0.000160448	-0.000255716	-0.00036137
mo	0,0013643	0,0029103	0,0046404	0,0065890
ts	0,0014774	0,0031518	0,0050254	0,0071357
als	0,4954921	1,0575446	1,6867826	2,3960360
br	-9.97596E-05	-0.000214044	-0.000341454	-0.000484428
nvk	0,0014513	0,0030978	0,0049395	0,0070137
svg	0,0079383	0,0170191	0,0272788	0,0389634
gr2	-8.13386E-05	-0.000173551	-0.000276662	-0.000391213
h1	-9.49315E-05	-0.000202886	-0.000323919	-0.000460647
nvk2	0,0014359	0,0030648	0,0048869	0,0069388
pg	-9.37395E-05	-0.000202534	-0.00032278	-0.000458616
svg2	0,0079492	0,0170424	0,0273162	0,0390170
hrs3	0,0013298	0,0028381	0,0045244	0,0064231
re	-7.93761E-05	-0.000169655	-0.00027094	-0.000385425
hrs	0,0013220	0,0028214	0,0044977	0,0063852
fr	-9.46403E-05	-0.000204195	-0.000325508	-0.000462561
strd	0,0639577	0,1371060	0,2197503	0,3138517
sd	0,0643367	0,1379170	0,2210477	0,3157008
ad	-8.88331E-05	-0.000189914	-0.000302795	-0.000428652
ev	0,0011764	0,0025096	0,0040013	0,0056813
fd	0,1143824	0,2446612	0,3911802	0,5571716
hs	0,0144124	0,0308974	0,0495227	0,0707324
v1	0,4936043	1,0535179	1,6803646	2,3869265
bodo	0,0013483	0,0028768	0,0045866	0,0065123

Table B.1: Change in node ranking values caused by simultaneous failure of links. The nodes the ranking values of which increase, they suffer overload, and might be a source of instability in the network.

Node	Δ(SLA vs 75%)	Δ(SLA vs 70%)	Δ(SLA vs 65%)	Δ(SLA vs 60%)
o1	-0.00053698	-0.000692607	-0.000873164	-0.001085157
o2	-0.0005251	-0.00067687	-0.000852728	-0.001058895
sto	-0.000570161	-0.000735322	-0.0009269	-0.001151806
o3	-0.000570134	-0.000735428	-0.000927161	-0.001151748
trd	0,0088391	0,0113818	0,0143213	0,0177582
tb	0,0095749	0,0123294	0,0155138	0,0192370
hb	0,0104532	0,0134609	0,0169383	0,0210047
sr	-0.00057653	-0.00074374	-0.000937811	-0.001165807
dr	-0.000633066	-0.000816987	-0.001030445	-0.001281274
krs	-0.000419962	-0.000540649	-0.000679926	-0.000842492
tos	0,0085699	0,0110332	0,0138800	0,0172091
kb	-0.000566672	-0.000729928	-0.000919194	-0.001143706
bo	-0.000576243	-0.000742658	-0.000935809	-0.001165222
brg	0,1277660	0,1657485	0,2103380	0,2633985
e1	-0.000576012	-0.000743075	-0.000936973	-0.001164759
as	-0.00057616	-0.000743263	-0.000937207	-0.001165052
dr2	-0.000669925	-0.000864603	-0.001090607	-0.001356363
tos2	0,0085290	0,0109803	0,0138131	0,0171256
tosS	0,0085269	0,0109775	0,0138096	0,0171213
brg3	0,4236649	0,5496024	0,6974408	0,8733580
tos3	0,0085309	0,0109827	0,0138161	0,0171294
bS	0,4236235	0,5495486	0,6973728	0,8732729
gr	-0.00048304	-0.000622207	-0.000783041	-0.000971124
mo	0,0087999	0,0113312	0,0142573	0,0176782
ts	0,0095302	0,0122716	0,0154407	0,0191459
als	3,2016287	4,1246589	5,1928821	6,4435703
br	-0.000649127	-0.000837401	-0.001055985	-0.001313952
nvk	0,0093678	0,0120565	0,0151706	0,0188119
svg	0,0523862	0,0679619	0,0862485	0,1080123
gr2	-0.000523072	-0.000673967	-0.000848481	-0.001052762
h1	-0.000616399	-0.000795278	-0.001002961	-0.001247109
nvk2	0,0092675	0,0119275	0,0150076	0,0186090
pg	-0.000615312	-0.000793172	-0.000999673	-0.001244875
svg2	0,0524583	0,0680556	0,0863676	0,1081618
hrs3	0,0085784	0,0110429	0,0138918	0,0172228
re	-0.000515891	-0.000665857	-0.000840105	-0.001045102
hrs	0,0085276	0,0109776	0,0138093	0,0171202
fr	-0.000620402	-0.000799842	-0.001008187	-0.001255273
strd	0,4219467	0,5473736	0,6946128	0,8698170
sd	0,4244266	0,5505820	0,6986717	0,8748817
ad	-0.000573516	-0.000739182	-0.000930993	-0.001156124
ev	0,0075873	0,0097694	0,0122916	0,0152400
fd	0,7467846	0,9654351	1,2203266	1,5212532
hs	0,0950966	0,1233682	0,1565584	0,1960562
v1	3,1894673	4,1090073	5,1732000	6,4191810
bodo	0,0086972	0,0111982	0,0140888	0,0174685

Table B.2: Change in node ranking values caused by simultaneous failure of links, when the failure rate is even higher, causing less than 75% availability of links.

Node number	Node Name	SLA ranking	Hypothetical ranking	Δ (%) SLA vs Hypoth.
1	ad	0,00016968	0,00018296	-7,83
2	als	0,00327250	0,00005813	98,22
3	as	0,01350305	0,01382890	-2,41
4	bo	0,01802466	0,01938051	-7,52
5	bodo	0,00009707	0,00000236	97,57
6	br	0,00213165	0,00229339	-7,59
7	brg	0,01570630	0,00006175	99,61
8	brg3	0,01022625	0,00004059	99,6
9	bS	0,00493377	0,00001966	99,6
10	dr	0,02631741	0,02820074	-7,16
11	dr2	0,01307523	0,01404941	-7,45
12	e1	0,01351874	0,01384316	-2,4
13	ev	0,00016935	0,00003569	78,93
14	fd	0,00001330	0,00000008	99,4
15	fr	0,00049596	0,00053163	-7,19
16	gr	0,00319219	0,00342350	-7,25
17	gr2	0,00154928	0,00166582	-7,52
18	h1	0,00132553	0,00136230	-2,77
19	hb	0,11913171	0,00296587	97,51
20	hrs	0,00052141	0,00001181	97,74
21	hrs3	0,00081057	0,00001915	97,64
22	hs	0,00004771	0,00002550	46,56
23	kb	0,01819442	0,01962817	-7,88
24	krs	0,02039188	0,02178066	-6,81
25	mo	0,00309482	0,00007688	97,52
26	nvk	0,00192340	0,00004321	97,75
27	nvk2	0,00100152	0,00002264	97,74
28	o1	0,56169878	0,57373840	-2,14
29	o2	0,55849933	0,57239090	-2,49
30	o3	0,17931679	0,18209094	-1,55
31	pg	0,00086967	0,00093746	-7,79
32	re	0,00065734	0,00066818	-1,65
33	sd	0,00023757	0,00000095	99,6
34	sr	0,02710624	0,02776485	-2,43
35	sto	0,53967585	0,55371575	-2,6
36	strd	0,00023808	0,00000156	99,34
37	svg	0,00181622	0,00109585	39,66
38	svg2	0,00087449	0,00052881	39,53
39	tb	0,11914965	0,00296658	97,51
40	tos	0,02003780	0,00048483	97,58
41	tos2	0,01067681	0,00025893	97,57
42	tos3	0,00966164	0,00023434	97,57
43	tosS	0,01065443	0,00025846	97,57
44	trd	0,12879274	0,00319153	97,52
45	ts	0,00286908	0,00007313	97,45
46	v1	0,00007890	0,00000140	98,22

Table B.3: **Hypothetical failures** - Network node ranking value results obtained after two main links(″o1″-″trd″ and ″o2″-″brg″) failures were analysed

Node number	Node Name	SLA ranking	February 2006	March 2006	Δ (%) SLA vs Febr.	Δ (%) SLA vs. March
1	ad	0,00016968	0,00017711	0,00016967	0,01	-4,38
2	als	0,00327250	0,00254390	0,00250118	23,57	22,26
3	as	0,01350305	0,01350709	0,01350387	-0,01	-0,03
4	bo	0,01802466	0,01892943	0,01800771	0,09	-5,02
5	bodo	0,00009707	0,00009526	0,00009699	0,08	1,87
6	br	0,00213165	0,00222893	0,00207148	2,82	-4,56
7	brg	0,01570630	0,01549627	0,01547953	1,44	1,34
8	brg3	0,01022625	0,00977331	0,00975366	4,62	4,43
9	bS	0,00493377	0,00471767	0,00470579	4,62	4,38
10	dr	0,02631741	0,02740775	0,02631545	0,01	-4,14
11	dr2	0,01307523	0,01362765	0,01307438	0,01	-4,22
12	e1	0,01351874	0,01352127	0,01351956	-0,01	-0,02
13	ev	0,00016935	0,00017064	0,00016923	0,07	-0,76
14	fd	0,00001330	0,00001239	0,00001233	7,32	6,83
15	fr	0,00049596	0,00051808	0,00049598	0	-4,46
16	gr	0,00319219	0,00332504	0,00319235	-0,01	-4,16
17	gr2	0,00154928	0,00161461	0,00154936	-0,01	-4,22
18	h1	0,00132553	0,00132784	0,00132531	0,02	-0,17
19	hb	0,11913171	0,12008342	0,11901185	0,1	-0,8
20	hrs	0,00052141	0,00047035	0,00052099	0,08	9,79
21	hrs3	0,00081057	0,00076368	0,00080991	0,08	5,78
22	hs	0,00004771	0,00004798	0,00004638	2,79	-0,56
23	kb	0,01819442	0,01910466	0,01819548	-0,01	-5
24	krs	0,02039188	0,02122293	0,02039274	0	-4,08
25	mo	0,00309482	0,00311857	0,00309221	0,08	-0,77
26	nvk	0,00192340	0,00174046	0,00192168	0,09	9,51
27	nvk2	0,00100152	0,00090963	0,00100064	0,09	9,18
28	o1	0,56169878	0,56151594	0,56172990	-0,01	0,03
29	o2	0,55849933	0,55829748	0,55853020	-0,01	0,04
30	o3	0,17931679	0,17755320	0,17932764	-0,01	0,98
31	pg	0,00086967	0,00091376	0,00086886	0,09	-5,07
32	re	0,00065734	0,00065786	0,00065738	-0,01	-0,08
33	sd	0,00023757	0,00022727	0,00022658	4,63	4,34
34	sr	0,02710624	0,02711809	0,02710136	0,02	-0,04
35	sto	0,53967585	0,53990927	0,53970837	-0,01	-0,04
36	strd	0,00023808	0,00022782	0,00022710	4,61	4,31
37	svg	0,00181622	0,00183647	0,00180458	0,64	-1,12
38	svg2	0,00087449	0,00088468	0,00086880	0,65	-1,17
39	tb	0,11914965	0,12010865	0,11904003	0,09	-0,8
40	tos	0,02003780	0,01942852	0,02002148	0,08	3,04
41	tos2	0,01067681	0,01035303	0,01066817	0,08	3,03
42	tos3	0,00966164	0,00937157	0,00965382	0,08	3
43	tosS	0,01065443	0,01033427	0,01064581	0,08	3
44	trd	0,12879274	0,12971716	0,12868322	0,09	-0,72
45	ts	0,00286908	0,00289365	0,00286646	0,09	-0,86
46	v1	0,00007890	0,00006141	0,00006035	23,5	22,16

Table B.4: **Live network analysis** - Network node ranking value results obtained after failures during February and March 2006 were analysed

Appendix C

Tools used

Graphs and pie charts were produced using GNUPlot Version 4.0 patchlevel 0 for Windows, and Sigmaplot 9.01, fully functional 30-day trial version.

Alternatively GNUPlot was used to obtain some of the graphs, using the script on a file ”test3.dat” which contains the columns with ranking values as shown in Table B.1:

```
set term postscript eps enhanced color
set output "graph.eps"
set key under
set xrange [0:46.2]
set yrange [-0.05:0.6]
set xlabel ’Node’
set ylabel ’Ranking value’
set grid
plot "test3.dat" using 1:3 title "New ranking"with lines, \
"test3.dat" using 1:4 title "SLA values" with lines
```

Some diagrams of networks were produced using Microsoft Visio, and templates for network architecture elements available in it.

Graphviz version 2.8 was used to produce the algorithm diagrams, and the fault tree. Graphviz script ”diagrams.dot” for the Figure 5.1:

```
digraph G
{
  subgraph cluster_0
  {
        edge [color=Blue];
        label = "Connectivity analysis";
        SLAC [ label="SLA connectivity \nvalue"];
        Cdown [ label="Observe link \ndowntimes"];
        Crec [ label="Recalculate \nconnectivity"];
        Cchange [ label="Observe change \nin connectivity"];
        Chigh [ label="High", shape=diamond];
        CInstab [ label="Instability suspected \ncheck ranking \nvalues",
                shape=box, style=rounded];

      SLAC -> Cdown;
      Cdown -> Crec;
      Crec -> Cchange;
      Cchange -> Chigh;
      Chigh -> CInstab [style=bold,label=" YES "];
      Chigh -> Cdown [style=bold,label=" NO "];
  }

  subgraph cluster_1
  {
        edge [color=Green];
        label = "Node ranking analysis";
        RLAC [ label="SLA ranking \nvalues"];
        Rdown [ label="Observe link \ndowntimes"];
        Rrec [ label="Recalculate \nranking values"];
        Rchange [ label="Observe change in \nnode ranking values"];
        Rhigh [ label="High", shape=diamond];
        RInstab [ label="Instability \nnoted", shape=box, style=rounded];

     RLAC -> Rdown;
     Rdown -> Rrec;
     Rrec -> Rchange;
     Rchange -> Rhigh;
     Rhigh -> RInstab[style=bold,label=" YES "];
     Rhigh -> Rdown [style=bold,label=" NO "];
  }

   {rank=same; SLAC; RLAC;}
   CInstab -> RLAC[style=dashed, color=red];
}
```

The script for the fault tree is the following:

```
graph ER
 {
   node [shape=box];
   A [label="Network Service \nFailure"]
   B [label="Software"]
   B1 [label="Routing \nProtocol"]
   B1a [label="Software \nbug"]
   B1b [label="Bad \nconfiguration"]
   C [label="Hardware"]
   C1 [label="Link"]
   C1a [label="Congested \nlink"]
   C1b [label="Damaged \nlink"]
   C2 [label="Node"]
   C2a [label="Interface \nfailure"]
   C2a1 [label="Power failure"]
   C2a2 [label="Node removal"]
   D [label="External"]
   D1 [label="Nuclear war"]
   D2 [label="Sabotage"]
   D3 [label="Cosmic forces"]
   E [label="Unknown"]

      A -- B
      A -- C
      A -- D
      A -- E
      C -- C1
      C -- C2
      D - D1
      D - D2
      D - D3
      B -- B1
      B1 -- B1a
      B1 -- B1b
      C2 -- C2a
      C2a -- C2a1
      C2a -- C2a2
      C1 -- C1a
      C1 -- C1b
      C1b -- C2a1
      C1b -- C2a
 }
```

The thesis was written in LaTeX language, more specifically in MikTex for Windows, and the GUI TeXnicCenter 1 Beta 6.31 (Firenze).