Final Thesis

A representation of Network Elements within the Operation Support Systems

by

Yannick Ceinos

LiTH-ISY-EX--05/3657--SE

2005-09-30

Examinor: Robert Forchheimer
Supervisor: Yann Le Bihan
**Titel**
Representation av nätverkeselment för nätverks-admintviua system

**Author**
Yannick Ceinos

**Abstract**
Currently, our world is getting more and more connected: information flows are everywhere at anytime. We are only at the beginning of the communication era. However, few want to know how those data are delivered as long as they are well delivered. Which paths they take, using optical fibres or regular copper cables, such questions do not matter for end-customers.

As a part of a telecommunication company, these questions do really matter. As networks become increasingly complex, it is significant to keep them under control and make them reliable to deliver high-quality services to users.

One of our main problems is how to get a representation of this network. More precisely how to get a relevant image of elements that compose this huge web and what points are significant in order to increase the quality of offered services. This thesis will try to answer these questions.

**Keywords**
Service Provider (SP), Operation Support Systems (OSS), router, Cascading Style Sheets (CSS), eXtended HyperText Markup Language (XHTML), Network Elements.
Upphovsrätt

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extra-ordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kräkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose.

Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law, the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/

© [2005, Yannick Ceinos]
Acknowledgements

This work completes a Master of Science in Computer Science: "Communication and Interactivity" degree at Linköpings Universitet, Sweden. This external thesis work has been realised within the Operation Support Systems department of a French Service Provider.

A thesis is a great opportunity to deepen a subject and to learn a lot. I would like to thank both my supervisor Yann Le Bihan and my examiner Robert Forchheimer for their advice, their supports and their relevant remarks regarding my work.

I also want to acknowledge my colleague Eric Alleaume for his long explanations about this complex network. Of course, I cannot forget to say thank you to all the others: Sandrine, Aymeric, David, Eric, Jean-François, Lionel and Xavier for the good atmosphere. Finally, I want to thank Michel Joubert for bringing to me the subject and giving me the opportunity to realise it.

My last grateful thank goes to my girlfriend Céline for the time and the help she brings to me.
Abstract

Currently, our world is getting more and more connected: information flows are everywhere at anytime. We are only at the beginning of the communication area. However, few wants to know how those data are delivered as long as they are well delivered. Which paths they take, using optical fibres or regular copper cables, such questions do not matter for end-customers.

As a part of a telecommunication company, these questions do really matter. As networks become increasingly complex, it is significant to keep them under control and make them reliable to delivery high-quality services to users.

One of our main problems is how to get a representation of this network. More precisely how to get a relevant image of elements that compose this huge web and what points are significant in order to increase the quality of offered services. This thesis will try to answer these questions.
# Table of Contents

1 **Introduction** .............................................................................................................. 1
   1.1 Motivation .................................................................................................................. 1
   1.2 Scope .......................................................................................................................... 2
   1.3 Study questions .......................................................................................................... 2
   1.4 Overview .................................................................................................................... 3

2 **Technical background** ............................................................................................... 5
   2.1 Some history .............................................................................................................. 5
   2.2 Global overview of an Autonomous System .............................................................. 7
   2.3 Zooming into one Autonomous System .................................................................. 9
      2.3.1 Network Elements and hierarchy ..................................................................... 9
      2.3.2 Communication within the Autonomous System ......................................... 11
      2.3.3 Routing and forwarding process ................................................................... 12
      2.3.4 Services offered by an Autonomous System ................................................. 13

3 **Guidelines** .................................................................................................................. 17
   3.1 What are the Operation Support Systems? ............................................................... 17
   3.2 Methodology ............................................................................................................. 19
      3.2.1 Choice of the data source ............................................................................... 20
      3.2.2 Elements that must be supervised ................................................................. 21
      3.2.3 How to retrieve and store the information? ..................................................... 26
      3.2.4 How to organise the data? ............................................................................. 29
   3.3 Presentation of dashboard ......................................................................................... 30

4 **SARIM: the tool** ......................................................................................................... 35
   4.1 Implementation .......................................................................................................... 35
   4.2 How pages are generated? ....................................................................................... 36
   4.3 Using Cascading Style Sheets techniques ............................................................... 37
      4.3.1 What are Cascading Style Sheets? ................................................................. 37
      4.3.2 Why using Cascading Style Sheets? .............................................................. 38
      4.3.3 A Cascading Style Sheets sample ................................................................. 39
   4.4 Screenshots of SARIM ............................................................................................. 43

5 **Conclusion** ................................................................................................................ 47
   5.1 Monitoring Network Elements is possible ............................................................... 47
   5.2 Using SARIM ............................................................................................................ 48
   5.3 Future work ................................................................................................................. 49

Appendix 1: Router configuration file .............................................................................. 54

Appendix 2: CVS file sample .......................................................................................... 56
Table of Pictures

Figure 1: Three stand-alone networks interconnected.........................................................6
Figure 2: The network of a Service Provider ........................................................................8
Figure 3: Methodology process ...............................................................................................20
Figure 4: Relationship between Network Elements ...............................................................25
Figure 5: Information flow from the network to database.......................................................26
Figure 6: Service viewpoint.....................................................................................................31
Figure 7: Technology viewpoint .............................................................................................32
Figure 8: Client viewpoint......................................................................................................32
Figure 9: XHTML file sample................................................................................................40
Figure 10: Block elements rendered without CSS file.............................................................41
Figure 11: CSS file sample......................................................................................................41
Figure 12: Block elements rendered with CSS file .................................................................42
Figure 13: Services distribution and router’s details.................................................................43
Figure 14: Interface viewpoint ...............................................................................................44
Figure 15: Services distribution...............................................................................................44
Figure 16: Client viewpoint ...................................................................................................45
Figure 17: Services detailed .................................................................................................45

Table 1: FCAPS management .............................................................................................19
# Abbreviations

This section contains all the acronyms used in this paper.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AF</td>
<td>Assured Forwarding</td>
</tr>
<tr>
<td>ARPA</td>
<td>Advanced Research Projects Agency</td>
</tr>
<tr>
<td>ARPANET</td>
<td>ARPA NETwork</td>
</tr>
<tr>
<td>AS</td>
<td>Autonomous System</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>BE</td>
<td>Best Effort</td>
</tr>
<tr>
<td>BGP</td>
<td>Border Gateway Protocol</td>
</tr>
<tr>
<td>CE</td>
<td>Customer Edge</td>
</tr>
<tr>
<td>CoS</td>
<td>Class of Service</td>
</tr>
<tr>
<td>CPE</td>
<td>Customer Provided Equipment or Customer Premise Equipment</td>
</tr>
<tr>
<td>CSNET</td>
<td>Computer Science NETwork</td>
</tr>
<tr>
<td>CSS</td>
<td>Cascading Style Sheets</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Value</td>
</tr>
<tr>
<td>DoD</td>
<td>U.S. Department of Defence</td>
</tr>
<tr>
<td>DSL</td>
<td>Digital Subscriber Line</td>
</tr>
<tr>
<td>EF</td>
<td>Expedited Forwarding</td>
</tr>
<tr>
<td>EGRP</td>
<td>Exterior Gateway Routing Protocol</td>
</tr>
<tr>
<td>FCAPS</td>
<td>Fault, Configuration, Accounting, Performance and Security</td>
</tr>
<tr>
<td>FR</td>
<td>Frame Relay</td>
</tr>
<tr>
<td>GPRS</td>
<td>General Packet Radio Service</td>
</tr>
<tr>
<td>HTML</td>
<td>HyperText Markup Language</td>
</tr>
<tr>
<td>IGRP</td>
<td>Interior Gateway Routing Protocol</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISDN</td>
<td>Integrated Services Digital Network</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunication Union</td>
</tr>
<tr>
<td>ITU-T</td>
<td>International Telecommunication Union – Telecommunication Standardization Sector</td>
</tr>
<tr>
<td>KB</td>
<td>KiloBytes</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MILNET</td>
<td>MILitary NETwork</td>
</tr>
<tr>
<td>MPLS</td>
<td>MultiProtocol Layer Switching</td>
</tr>
<tr>
<td>NCP</td>
<td>Network Control Protocol</td>
</tr>
<tr>
<td>NE</td>
<td>Network Element</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection</td>
</tr>
<tr>
<td>OSPF</td>
<td>Open Shortest Path First</td>
</tr>
<tr>
<td>OSS</td>
<td>Operation Support Systems</td>
</tr>
<tr>
<td>P</td>
<td>Provider</td>
</tr>
<tr>
<td>PE</td>
<td>Provider Edge</td>
</tr>
<tr>
<td>POS</td>
<td>Packet Over SDH / SONET</td>
</tr>
<tr>
<td>PPP</td>
<td>Point-to-Point Protocol</td>
</tr>
<tr>
<td>PVC</td>
<td>Permanent Virtual Circuit</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RIPv2</td>
<td>Routing Information Protocol version 2</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SDH</td>
<td>Synchronous Digital Hierarchy</td>
</tr>
<tr>
<td>SONET</td>
<td>Synchronous Optical NETwork</td>
</tr>
<tr>
<td>SP</td>
<td>Service Provider</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TMN</td>
<td>Telecommunication Management Network</td>
</tr>
<tr>
<td>VC</td>
<td>Virtual Circuit</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>VPN SC</td>
<td>VPN Solution Center</td>
</tr>
<tr>
<td>VRF</td>
<td>VPN Routing and Forwarding</td>
</tr>
<tr>
<td>XHTML</td>
<td>eXtensible HyperText Markup Language</td>
</tr>
</tbody>
</table>
1 Introduction

This thesis work is the second and final part of a two-part project about representing Network Elements in the Operation Support Systems department of a Service Provider. Conclusions of the first piece of work show that retrieving data from such a network is possible. Now, the second part, based on the former results, is focused on getting reliable data and modelling them in a specific representation.

First, this section gives motivation regarding the thesis subject before setting up its scope. Then, it states the questions that lead the thesis. Finally, it gives the reader a global overview of the thesis contents.

1.1 Motivation

In our world, we are getting more and more connected to each other. Mobile phones, computer networks and in the immediate future, it is easily conceivable to have intelligent devices linked to each other, chatting freely together. For instance, automatic filling out fridge, automatic-driving cars avoiding traffic jams, etc… This enumeration is a very good example of what is a communicating world. To support all this sum of data flows, powerful and huge networks are needed. However, as the amount of devices connected raises quickly, the network complexity increases as well and the network interactions are numerous.

Over those networks, Telecommunication Service Providers supply services. One of the main requirements is to understand what happens in their own network. They want to enhance their knowledge of what it is composed by, what elements are impacted when a part of the network fails, etc…
This thesis subject is born out of the desire of a better understanding of those interconnected networks. The work is monitoring and modelling oriented. The challenge of the thesis is to represent data from the network in such a way that managers and operational engineers quickly retrieve data and take appropriate decisions based on those data.

1.2 Scope

This study is focused on the elements composing the network, the so-called Network Elements (NE), and a way of representing them for users. Several layers exist in the network representation; here the point of view adopted is the one just at the border between the Network Element’s layer and the Operation Support Systems’ layer. The first one is hardware related while the second one is more focused on the services runs on this hardware.

Security matters, dimensioning, provisioning, routing and forwarding problems are not in the scope of the thesis. Nevertheless, some references are made for a better comprehension of the thesis background. Various documentation [1], [2] (security) and [3] (architecture to provide the studied network) are recommended readings for those concerned.

1.3 Study questions

This thesis has one main goal namely to find a representation of the Network Elements. As the term representation is vague, it needs some complementary explanations. In order to provide modelling, this thesis needs to answer three sub-questions.

- **What Network Elements can be monitored within the network of a service provider?**

If this question finds out an answer then, the thesis can be pushed one step further and another question raises.

- **Is it possible to retrieve monitoring data? If so, how to get them?**
Given that data which fully filled the requirements are found and it is possible to retrieve them from the network. The next question is.

- **How to combine them such that they are useful?**

The thesis will answer these questions, emphasising the answers on the background needed and the technical solutions found.

## 1.4 Overview

This section gives readers a brief overview regarding the thesis' contents.

*Chapter 1 - Introduction* explains briefly what are the aims of the thesis, which elements have motivated this study and how it was proceeded.

*Chapter 2 - Technical background* gives technical tools and knowledge to understand the purposes discussed here.

*Chapter 3 – Guidelines* focused on the methodology used to answer the questions raised in the Introduction. The chapter provides a logical solution.

*Chapter 4 – The tool SARIM* explains the work done during the thesis detailing the technologies used and showing the obtained result.

*Chapter 5 - Conclusion* concludes this thesis showing what have been done and what is needed in order to improve the provided solution.
2 Technical background

This section gives technical knowledge for a better understanding of the achievements that have been done during this thesis work. For someone who does not have a strong background in the network field, this section is answering to the following question: "What information do I need to know to understand this thesis?"

First, this section starts explaining the birth of the Internet in order to understand its heterogeneity. It also introduces the notion of stand-alone networks called Autonomous System. Finally, the way an Autonomous System works is detailed in the last part explaining both how it works and what it is composed by.

2.1 Some history

After the Second World War, the world entered into another major conflict "The Cold War" which gave birth to two opposed blocks pro-American versus pro-Soviet. Competition in the space run and in nuclear control had begun. Concerned by this threat, the U.S Department of Defence (DoD) created a special section in charge of developing sciences and new technologies with military applications: the Advanced Research Projects Agency (ARPA).

One of ARPA's main projects was to create a communication network that could endure a nuclear strike and still work even if one or several nodes were destroyed. In 1968, only six years after its creation, such a network ARPANET is born. The first physical lines are deployed the year after, connecting four American universities.

During the next decades, the former small network has grown rapidly giving birth to one military network MILNET and to some specific interconnected networks such as ARPANET (the non-
military part of the old ARPANET), CSNET (the network of the National Science Foundation), etc… All hosts are connected together through one protocol\(^1\) and all networks are interconnected together through special protocols\(^2\). All those networks are a part of the actual Internet.

![Diagram of interconnected networks](image)

**Figure 1: Three stand-alone networks interconnected**

The history of Internet is not in the scope of this thesis and is just here to explain the heterogeneity and the structure of the interconnected networks. As during the evolution of the former ARPANET into the Internet, several networks were born simultaneously. All these networks are, for sure connected together. However, they can perfectly work as stand-alone networks. This set of hosts is called Autonomous Systems (AS).

For further information about Internet history, the reader is recommended to have a look at [4] and [5] that give some detailed information about the Internet creation.

---

\(^1\) Since 1983, the Transmission Control Protocol / Internet Protocol (TCP / IP) replaces the old Network Control Protocol (NCP).

\(^2\) The Border Gateway Protocol is usually used to interconnect networks.
2.2 Global overview of an Autonomous System

Now that the topology of the interconnected networks is clearer, let us focus on one of these ASs. According to [6], an AS is defined as follow:

"[...] an Autonomous System is a set of routers under a single technical administration, using an interior gateway protocol and common metrics to route packets within the AS and using an exterior gateway protocol to route packets to other AS's."

RFC1772 – 1. Introduction

Data transmitted through the network are lumped into packets. They go from their source to their destination in a connectionless way. This signifies that they contain enough information to be routed from one node to another independently from each other. At each node the packet stops by, the router opens it, looks at the packet’s header and takes a routing decision based on that information. Routing algorithms chose the best route, best has several meanings: the shortest in time, the straightest (not a lot of hops), etc… Routers regularly exchange routing information about their neighbours using an Interior Gateway Routing Protocol (IGRP) such as Routing Information Protocol (RIP).

At the routing layer and defined in a simple manner, an AS is represented by a number used by routing protocols to avoid routing loops. From a physical point of view, several elements, such as routers, wires, servers, clients, etc… build a stand-alone telecommunication network. The company within which this work is done, uses a network-based Internet Protocol - Virtual Private Network over MultiProtocol Layer Switching (IP-VPN over MPLS). It means that it implements VPN using the MPLS protocol.

The firm is a Telecommunication Service Provider (Service Provider or SP) and owns the AS.

The following diagram is a relevant example of such a network. It is a simplification of a schema shown in [7]. The schema shows how relationships between all NEs are established in a
network-based IP-VPN over MPLS (see section 2.3.3 for explanations regarding MPLS). The details of the diagram will be detailed below.

![Network Diagram]

**Figure 2: The network of a Service Provider**

**How information flows from one AS to another?**

Even if ASs work perfectly as stand-alone networks, they may sign up agreements to route foreign traffic through their network. Several web-pages\(^3\) are available that show agreements contracted between one AS and its neighbours. Routers exchange route information through an Exterior Gateway Routing Protocol (EGRP) such as the Border Gateway Protocol (BGP).

\(^3\) RIPE NCC’s whois web-page http://www.ripe.net/perl/whois
2.3 Zooming into one Autonomous System

This section gives details about the Network Elements organisation and their connections.

Note: Here, only a simple AS is represented. Complex examples of ASs can be found e.g. in [18].

2.3.1 Network Elements and hierarchy

The network core, which is also known as the backbone, is composed of powerful routers (e.g. routers having strong computing capabilities) called Provider routers (P routers). The aim of those routers is to send data, through the network, as fast as possible.

Within the backbone range and directly connected to one or several P routers, there are the Provider Edge routers (PE routers). This multi-connectivity allows redundancy in case of one of the P routers going down. It also permits direct communication with the most appropriate P router. From a provider point of view, they are the last layer of the SP's network. As far as they are concerned, they are acting as a kind of buffer between the SP's network and the customers' devices.

Finally, on the customer’s side, there are the Customers Edge routers (CE router) or CPE which stands for Customer Provided Equipment or Customer Premise Equipment. Generally, these routers belong to customers. They provide a customer access point to the SP’s network. As for the PE routers' connection, they are directly connected to one or several PE routers.

The SP is in charge of managing the P routers and the PE routers whereas customers, generally, take care of the CE routers (except if customers use the SP to manage their devices). In the company where this work is done, the CEs are managed by the SP. This is included in the commercial package.
Other elements appearing in the picture are **Virtual Private Network** (VPN) and **sites**. One is implemented inside the other one; a site always belongs to one or several VPNs. This concept is clearer and more relevant with an example.

Consider a company that has several subsidiaries scattered around the world and wants to connect them all together (i.e. build a Private Network). Instead of building its own infrastructure (e.g. network, routers, etc…) which is very expensive, the firm asks a SP to provide this kind of structure. This is shared with other customers but there is no interaction between customers data flows (i.e. they remain private). From a MPLS point of view, these subsidiaries are called sites and are attached to the SP’s backbone via CE routers. If this set is divided into subsets, the resulting subsets are called VPNs. The customer usually defines a set of policies that rules the VPN (how sites “talk” to each other, etc…). Finally, the SP sets up his devices in order to take into account the customer’s requirements. Other services may be implemented such as security policy, QoS management, remote accesses, etc…

To give some quantitative data about the routers' repartition, for a dozen of P routers, there are around a hundred of PE routers and more than thousands of CE routers. Concerning the sites and the VPNs, it is very difficult to give numbers because one client may have ten sites and one VPN whereas another one may have four sites and three VPNs. These two last factors are up to customers. Generally, one customer has one VPN composed from at least two up to thousand sites.

The dimensioning parameters of an AS (network-based IP-VPN over MPLS) depend on:

- the expected number of customers,
- the expected number of VPN,
- the expected number of sites per customer.
2.3.2 Communication within the Autonomous System

Not all components of an AS communicate directly together; the following list shows the occurring combinations:

- communication between two P routers,
- communication between a P router and a PE router,
- communication between a PE router and a CE router,
- communication between two sites.

Those are the only possible ways of communication.

Communication between two P routers

Generally, the P routers are connected together using some optical fibres. The protocols used to send data over optical fibres is the Synchronous Digital Hierarchy (SDH) and the Synchronous Optical NETwork (SONET), they are respectively the International standard and the United States standard. The International Telecommunication Union - Telecommunication Standardization Sector (ITU-T) charters specifications for this recommendation in [8]. This is the faster link within the backbone, as P routers are used to send data as fast as they can. The technology used to connect them is designed to support high data transmission rate.

Communication between a P router and a PE router

PE routers are linked to one or several P routers. As for P routers, they must send data as fast as they can so they also use high transmission rate technologies. Optical fibre devices are expensive and PE routers are more numerous than P routers. The technologies used are Fast Ethernet or Packet over SDH / SONET (a.k.a. POS) are used.

Communication between a PE router and a CE router

CE routers are customer entry points to the SP's network. As clients do not have the same needs, a wide range of technologies is available for the physical PE – CE link. Asynchronous Transfer
Mode (ATM), Frame-Relay (FR) or snet are example of used protocols. For instance, it may use Fast Ethernet as well, if it needs high bandwidth whereas if it only needs a permanent connection with low bandwidth, xDSL (X Digital Subscribe Line) can be used. Non permanent accesses such as Integrated Services Digital Network (ISDN) or General Packet Radio Service (GPRS) may be used as well.

**Communication between two sites**

Sites are not directly connected to each other, so communication must go through a set of devices (e.g. P, PE, and CE routers) within the SP’s network. Moreover, two sites communicate only if they belong to the same VPN. There are two ways of topology. The communication can be “Full Mesh”, meaning that all the sites within the VPN speak to each other. Alternatively, the communication can be “Hub and Spoke”. One site is the Hub and all the others are the Spokes, communication is only Hub to Spokes or vice-versa. Meaning that this is a client / server relationship.

**2.3.3 Routing and forwarding process**

The SP runs a network-based IP-VPN / MPLS, thus the routing and forwarding process within the backbone is based on the MPLS protocol. It has been chartered and standardised in several RFCs by the Internet Engineering Task Force⁴ (IETF); [9] describes the architecture of the network whereas [10] gives an example on how to implement this technology.

First of all, Cisco Systems Inc.⁵ has invented a method called “Tag Switching” which is the basis of the network-based IP-VPN / MPLS implementation. It uses labels to combine two different techniques, Virtual Circuit (VC) and datagram. A VC is a logical path between nodes, it can be permanent or switched. A datagram is a package that contains enough information to be routed in a connectionless network service to it destination. This is a synonym of a packet.

---

⁴ IETF's homepage http://www.ietf.org
MPLS protocol assigns labels to packets at the ingress node (i.e. the entry point). They indicate the path to follow through the network. Routing decisions are based on the label. There is no need to read the packet header, so routing decisions are taken faster. Each router (e.g. P, PE and CE routers) has a VPN Routing and Forwarding table (VRF). It contains routing information such as topology of the network, next hop, etc... Routers look up in their VRF to match the appropriate label and forward the packet to the next hop.

In [10], section 3 details the labelling and forwarding processes.

Moreover, for a better way of congestion handling, Differentiated Services capabilities may be added to the routing process (see next section 2.3.4.2 for details).

### 2.3.4 Services offered by an Autonomous System

The main goal of a SP is to offer services to customers such as delivering packets until they reach their destination, supplying good Quality of Service (QoS), etc... This section puts the emphasise on two services offered by the company where this work is done.

#### 2.3.4.1 VPN

VPNs allow a customer to connect several sites together. From the customer's point of view, a VPN is more or less, like its own network except those hosts are generally on the different geographical location.

In our case security over VPN is reliable and fine. Indeed, packets use the SP’s network and not the public Internet. This is how VPNs are implemented in the company. Hence, the only way to

---

5 Cisco Systems Inc.'s homepage http://www.cisco.com
eavesdrop data flows is to hack directly the SP’s network or facilities which is something more difficult that sniffing encrypted packets over the Internet.

2.3.4.2 Prioritised flows

The services management over the network is a major part in the SP's scope of activities. It allows:

- packet classification,
- congestion avoidance,
- congestion management.

Packet classification adds label to packets in order to sort them. Congestion management sets policy to deal with congestion when it occurs whereas congestion avoidance tries to prevent congestion from happening.

One method of managing packets is called Differentiated Services capabilities (DiffServ) and is implemented over the network own by a SP. A set of rules defines guidelines for packets. Routers within the SP's network implement the DiffServ features. The most widespread implemented scheme is to set up DiffServ capabilities only over the PE – CE links. Moreover, the rest of the network (i.e. the backbone) is over dimensioned. The limitation is due to the youth of the technology. Algorithms are not reliable and powerful enough for a full implementation over the SP’s network.

DiffServ capabilities are implemented via several Classes of Services (CoS) which are:

- Expedited Forwarding class (EF class),
- Assured Forwarding class (AF class),
- Best Effort class (BE class).


**Expedited Forwarding class**

It is defined in [11] and completed by [12], the following sums up the aim of this CoS type.

"The Expedited Forwarding (EF) Per-Hop Behavior (PHB) was designed to be used to build a low-loss, low-latency, low-jitter, assured bandwidth service"

RFC 3247 – 1. Introduction

When a packet is forwarded using this policy, it must be done in the time due, independently from the amount of packets that have arrived or that have left the device.

**Assured Forwarding class**

It is defined in [13], the following sums up the aim of this type of CoS.

"In a typical application, a company [...] wants an assurance that IP packets within this intranet are forwarded with high probability as long as the aggregate traffic from each site does not exceed the subscribed information rate (profile)."

RFC 2597 - 1. Purpose and overview

AF class is divided into 4 sub-classes, each sub-class has its own characteristics. It simply means that for each class, an amount of forwarding resources (buffer, bandwidth, etc…) is allocated. In addition, a drop precedence level is set. In case of congestion, packets having the highest precedence level are dropped. None-used resources by AF classes may be reallocated to other classes. This is a complex mechanism defined in [13].

**Best Effort class**

No rules are set along the packet’s road, devices try their best to deliver them. If congestion happens, then packets might be discarded or queued according to the set of policies chosen.
To sum up, EF is the highest prioritised flow and it is required by real-time applications such as voice, videoconference, etc… Delays are not possible because applications can not stand delays. If voice packets arrived with a long delay, a part of the conversation is cut off and this is a serious problem.

Then AF and BE classes try their best. AF classes get the highest priority after EF classes. AF flows are usually reserved for critical applications (e.g. Client / server applications, ERP, etc…), and BE for standard applications (e.g. HTTP, mail, etc…).

**How do customer choose CoS?**

Customer do not choose CoS directly. In fact, after auditing their needs, the SP provides packs that are especially designed to fit their needs. To make those features work, the SP has to set up routers with the right parameters.
3 Guidelines

The former section presented the technical background. The main issue discussed in this section is the theoretical part of my own work with the company.

First, this chapter introduces the position of the Operation Support Systems department in monitoring the SP’s network. Then, it explains the methodology used to answer the study questions. Finally, it describes the logical representation of the NEs.

3.1 What are the Operation Support Systems?

Let me start this section with a sentence quoted from a Cisco Systems Inc.’s white paper [14]

“The goals of a telecommunications service provider are to quickly provision and provide revenue-producing service; assure that the revenue-producing network remains available to provide quality service, and measure service usage or network availability for efficient and effective billing and revenue recognition.”

Introduction

In other words, once the network is set up, the first priority is to keep the network downtime to a minimum in order to provide high-performance services over the network. All SPs use more or less the same technology, differences between SPs reside into the management of their network and the way they provide services.

To achieve these goals, a SP must have a Telecommunication Management Network (TMN) methodology as the ITU-T described it in a set of documentation. Even if those documents were originally made for telecommunication networks, they can easily be applied to data networks. To meet those requirements, SP usually implements at least, the FCAPS management. The
Operation Support Systems (OSS) department plays a major role in this achievement because it is in charge of applying this management. FCAPS is an acronym for:

- Fault management (monitoring the network, check for alarms, etc…),
- Configuration management (provisioning, setting of the machines, etc…),
- Accounting management (billing),
- Performance Management (statistics about devices),
- Security (set of security rules to get in touch with the network).

Fault management

Fault management detects, logs, notifies users and tries to fix “small” problems. This technique is the simplest and most widely spread within the FCAPS method. All the problems encountered and the solutions found are stored in a pseudo-database. It is there to provide already-tested solutions when a problem re-occurs.

Configuration management

Configuration management handles all operations that install or remove materials. It also takes care of monitoring the changes of parameters.

Accounting management

Accounting management measures network’s characteristics (e.g. bandwidth, traffic generated, etc…) in order to assure the billing process. For instance, it permits to compare what has been sold to a customer and what this customer really uses.

Performance Management

Performance management analyses network performance and determines thresholds to trigger alarms. Data transmission rates, CPU resources, etc… are the kind of data monitored.
Security

Security management controls access to the SP’s network according to a set of rules. It acts to ensure that no damage is made against the network.

Each management block deals with a set of functions, the following table sums up actions that can be realised. It is extracted from [15], section 3.

<table>
<thead>
<tr>
<th>Fault Management</th>
<th>Configuration Management</th>
<th>Accounting Management</th>
<th>Performance Management</th>
<th>Security Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alarm handling</td>
<td>System turn-up</td>
<td>Track service usage</td>
<td>Data collection</td>
<td>Control NE access</td>
</tr>
<tr>
<td>Trouble detection</td>
<td>Network provisioning</td>
<td>Bill for services</td>
<td>Report generation</td>
<td>Enable NE functions</td>
</tr>
<tr>
<td>Trouble correction</td>
<td>Auto-discovery</td>
<td></td>
<td>Data analysis</td>
<td>Access logs</td>
</tr>
<tr>
<td>Test and acceptance</td>
<td>Back up and restore</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network recovery</td>
<td>Database handling</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: FCAPS management

The tools designed for the OSS department will implement some of the FCAPS requirements such as Fault, Configuration, Accounting management. Not all the sub-functions but just some of them are implemented. The tool is able to detect configuration mistakes within the configuration of an interface or a sub-interface. It gets its information from the network so in a way it knows how it is configured in the network. Finally, it has been designed for counting, it is not used for the billing process but it gathers and counts data that belong to the same categories (e.g. interfaces, routers, etc…). All those functions are detailed in the following sections.

3.2 Methodology

The tools get data from the network and makes them available for users in the form of dashboards. Which data must be monitored? How should they be represented? This section is
going to answer these questions. The following figure illustrates the different steps to achieve those goals.

![Figure 3: Methodology process](image)

**3.2.1 Choice of the data source**

Other ways of representing the network exist. This section details them and presents what it is expected by introducing a new representation. The following are the common way of representing the network:

- statistics,
- repository.

**Statistics**

At the OSS department, there are several tools that monitor the network. Here, ‘monitor’ means counting. Given the IP address of a router, it creates graphics about the bandwidth currently used by an interface. For example every 5 minutes, the tool counts how many sessions are opened, how much bandwidth is used on this interface, etc… These measures are used by the billing service in order to provide invoices to the end-customers.
These tools also play another role. They trigger alarms when elements in the network go wrong. For instance, if an interface goes down for several minutes, an alarm is launched. Hence, engineers find the problem, determine what impacts this causes on the network before fixing it.

Repository

The other way of representing the network is using a repository. This is a database where all the devices that have been ordered, installed and removed are stored. Every change made in the repository must be reflected in the SP’s network.

Statistics are perfect when counting how much resources an element is consuming. But, here, the goal is to understand what the network is composed by. Thus, using the repository as a source of data could be a relevant choice. Except that there is a gap between what has been ordered and what it is in the network. For example, a client has resigned. All its resources are removed from the repository but few remain due to some problems. The repository does not reflect the network.

Hence, the best data source to obtain the representation of a network is the network itself. The following sections from chapter 3 explain the steps taken to achieve this goal.

3.2.2 Elements that must be supervised

The network of the SP is complex and several different NEs are involved in the delivery of packets. Every single element inside the network can be counted and represented but some choices are more relevant than others. Let us see what are those elements in the scope of monitored elements.

Routers are the top-acting elements in the forwarding process. So all the monitoring data will come from this source of information. Three types of routers compose the SP’s network. Studying all the routers together is useless because adding data from one type of router to another do not have interest. Hence, only one type of routers is kept.
The following arguments indicate that P and CE routers are not good candidates for being our source of data. P routers do not implement services because they are there to send data as fast as possible. They know nothing about what they are delivering. On the other hand, CE routers have information about services but they are usually managed by the customers. In this case, this is difficult, for a SP, to access the routers and to retrieve information from them. Moreover, as CE routers are too numerous, monitoring traffic over the network is huge and would disturb the network (this purpose is discussed in section 3.2.2).

Finally, only one kind of router remains namely PE routers and they have two main advantages. First, they are buffers between the customers layer and the core layer, so they are in touch with every single piece of hardware within the SP’s network. Information they hold is representative from the network. Second, they are not too numerous, so collecting data is possible and do not congest the network.

Now that the scope is reduced to one kind of router, several other NEs remain. Some are physical such as interfaces some are logical as for instance VPN Routing and Forwarding (VRF). The following are the elements that are going to be monitored. For each, an example of what is expected is given.

- * hostname

  This is the name of the router. In order to be consistent, the name fits normalisation rules designed by the company engineer architecture. For instance, the first three letters stand for the location, etc… Example: VALIMC7501

- * interface

  This is the name of the physical device installed into the router. As for the hostname, it fits normalisation rules. Example: Serial5/0/0

- * sub-interface

  When an interface is divided into logical ones, these subdivisions are called sub-interfaces. This is a number is concatenated to the interface name. Examples: 123, 234, etc...
- * interface type

An interface is generally divided in virtual subdivisions. This field indicates if the device is a real interface or a logical division. The values are PHYSICAL and LOGICAL.

- * administrative status

This indicates whether the interface is UP (enabled) or DOWN (disable). The values are UP and DOWN.

- * Permanent Virtual Circuit * (PVC)

This indicates the virtual circuit that borrow the packets emitted and received on this device. Example: (1/450)

- bandwidth

This is the transmission data rate affected to this device. It is expressed in kilobytes per second. Example: 40000

- client

This is the name of the client that this interface “belongs to”. Example: Linköping University

- VRF

This contains the name of the number VPN that this device is attached to. Example: V000:LINKOPING-UNIVERSITY

- * rt_import

This gives the names of the incoming routes and specifies the name of the AS they belong to (first part of the sequence). Example: /000:345

- * rt_export

This is the same as for rt_import but for outgoing routes. Example: /000:234

- Pack QoS

This defined the Quality of Service used for this device (i.e. the type of service subscribed). Example: PACK-DATA
- technology

This indicates the OSI layer 2 name. Example: AAL5SNAP, PPP, FR

- * description

This indicates complementary information. This is a textual label. Example: Provisioned by VPNSC, Service Request #345

- * IP address

This gives the IP address to access the device. IP Address can be public or private as described in [21]. Example: 172.30.0.10

There are two kinds of data. On one hand, there is the informative part, it means that the data will be gathered and counted under one category. On the other hand, there are attributes, they are here to provide additional information but they are not taken into account while merging the data. Above, those data are marked with a star before their name.

The following diagram shows the relationship between all those NEs. The numbers indicate the cardinality of the relationship.

Note: There are two configurations possible, either an interface has a sub-interface or not. If there is, at least, one sub-interface, then it is linked to the VRF. If not this is the interface which is connected to the VRF.
**Why are these elements selected?**

In an AS, the essential elements are routers; they are “the hosts” of the other elements. All the other elements are attached or linked to the router. Thus, the first router-related elements are **interfaces**. They are physical cards plugged into the router. They contain **sub-interfaces**, which are logical sub-divisions of one interface. For instance, an ATM interface is divided into fifteen ATM sub-interfaces. Those elements give information about the router capacity to host new cards or logical sub-divisions. These interfaces or sub-interfaces are accessible through an **IP address**.

Then, the interface **technology** gives indications about the other parameters. An ATM interface does not have the same configuration parameters as a FRAME-RELAY interface does.

**VRF** shows two interesting details. First, the number of VPN available within the network, an important data for capacity planning and network dimensioning. Secondly, VRFs show the VPN the client belongs to.

**Client** names are monitored as well. For example, it is interesting to know the names of the client that are impacted when a router goes down. The sites distribution between the different
routers is also a pertinent information. Having all the company’s sites on one unique router is definitively not good if the router goes down.

**Bandwidth** associated with the technology and **pack QoS** are relevant to understand what are the trends within the SP’s network. Which is the most sold bandwidth? Which are the most implemented packs? This kind of data helps marketing division to supply commercial offers that are more consistent.

Other elements exist in the network such as service requests; they set up resources on the routers. However, they will not be monitored on the system because they are at a higher level of conception and this study is focused on NEs’ point of view.

### 3.2.3 How to retrieve and store the information?

The following diagram illustrates the different steps taken in the gathering of the information. Explanations are given below.

![Figure 5: Information flow from the network to database](image)

**Step 1: Retrieving information**

Information can come from several sources (e.g. network, client’s ordering, client’s delivering, purchases department, etc…). In my case, they only come from the SP’s network because the goal is to know what is provisioned over the network, not what should to be provisioned. Moreover, using one source of information avoids inconsistency into the data set.
Thus, monitoring data come from a unique source: the SP’s network. Nevertheless, two tools are used to retrieve those information. One is VPN Solutions Center\(^6\) (VPN SC) the other one is a custom-made PERL\(^7\) script.

The main part of collecting data is handled by the PERL script which examines each router configuration file (i.e. a text file). It contains information regarding the different parameters that can be set up on a router and some additional parameters concerning the routing process. As for instance, the name of an interface, its bandwidth, its sub-interfaces, VRFs, etc… (c.f. Appendix I to get a configuration file sample).

The script recognises and extracts some specific parameters using regular expressions and other parsing functions. The piece of code is not very tolerant because it recognises only what it is supposed to identify. Nevertheless, it is modular enough to add new detection functions taking into account new technologies.

Finally, the second part of the retrieving process is handled by VPN SC. There is only one missing data that cannot be collecting using the configuration file: the client. VPN SC has the missing information about clients. It holds records that associate a VRF with the client it is attached to.

Retrieving data from the network generates additional traffic. In the case of my tool, information is extracted from the routers configuration files. Tools exist and keep a trace of those files. VPN SC is one of these tools. Everyday it is set up to store a copy of these configuration files in a directory. The PERL script lists the files contained in the directory and then starts parsing each of them. To give some numerical data, a configuration file of a PE router is about 300KB (up to 400KB for the most provisioned). VPN SC uses the router management port to download those files. Compared to the traffic that goes through this router, the amount of transmitted data is unimportant.
Step 2: Storing information

Information are stored in a Comma Separated Variable (CSV) file. First, the original file is stored on the VPN SC server before being copied on the server that runs my tool. Each row of the file contains data about one interface or one sub-interface (c.f. Appendix 2 shows some rows extracted for one of the CSV file). The different fields are the ones mentioned in section 3.2.1.

The CSV file size is about 4.25MB and contains around 18000 lines. It keeps growing so a compression process will be added in order not to use too much resources.

The CSV file format is very useful to store data because of it simplicity. Data are stored and separated by commas. However, its limitations come from the format itself. Retrieving simple information from a CSV file is easy. It just needs to apply text filters on the desired columns. When it comes to gather a huge amount of data with the help of regular expressions, the querying process becomes a tough work. That is why data are imported into a database, which provides structured information and a better way of retrieving data. Moreover, access times are lower and faster.

The entire CSV files are kept to have a trace of the original data because only gathered data are stored into the database. Another reason is that the tool has been designed to act in a pre-programmed style, people might be interested in calculating something new. This file is more flexible than the database.

---

6 This is an OSS made by CISCO Systems Inc. in order to help the VPNs configuration.
7 Homepage www.perl.org
3.2.4 How to organise the data?

Note: Records in the CSV file contain both interface and sub-interface parameters, to be rigorous I should always use both terms but in order not to overwhelm the text, I may use only one of those terms.

Now that all the requirements are respected, data need to be gathered and counted under one category using common parameters. The counting process needs to be accurate; otherwise, this is pointless. Thus, the first step is removing records that are mis-configured so they are not taken into account while the tool is counting. There are two types of invalid records: errors and warnings. Errors are typically interfaces or sub-interfaces that cannot be summed because their configuration parameters cannot work for real. On the other hand, warnings are generally typing mistakes that may perturb the accounting process but not always. More important, they do not disturb the function of the router.

One of the biggest problems, is the data reliability: counting mis-configured interfaces is not an obvious process. When counting objects in a strict way, some interfaces might be forgotten because they do not fit in the requirements. The calculated numbers are right because the program sticks to the rules. Nonetheless, these data are also wrong because they do not represent what there really is in the network. It has to tolerate some mistakes. Here the solution is to add a special field saying how many interfaces are mis-configured (errors) or containing typing mistakes (warnings).

Here is an example to make this clearer. Suppose an interface shows a bandwidth equal to 608 while the right number is 607. If the algorithm sticks to the set of defined rules, this interface is not counted even if resources are affected and traffic is generated on the interface, which means that it is working perfectly well. Therefore, it must be counted to give users a more accurate representation of the network. The difficulty is to define rules, which tolerate only working interfaces and not everything.
Unfortunately, there is no specific way of finding those rules. Network architecture engineers write provisioning rules but sometimes, mistakes slide during the set-up. Making the tool more accurate is based on facts. Considering one specific client, it should have 200 sites and there are only 150. Something goes wrong. After investigation, together with the help of the network architecture engineers, we found that there was a typing mistake in the bandwidth parameter. So a new rule has been added to the error detection process.

Results of the gathering process are stored into the database, which give more flexibility to retrieve data

### 3.3 Presentation of dashboard

As is noticed in the former section, data are different and can be classified under various categories. Here, they are represented by dashboard, which is a hierarchical structure that gathers information under common characteristics. The classification depends on the elements. Naturally, several categories arise and each one is a kind of viewpoint, which is:

- services viewpoint,
- technology viewpoint,
- client viewpoint,
- regular information about router and interface,
- history.

This section gives a logical representation of the dashboard whereas section 4.4 illustrates those purposes with screenshots from the tool.

**Service viewpoint**

The service viewpoint is focused on the pack QoSs that are sold. Figure 7 is a logical representation of the data organisation in this dashboard. Considering the “access points” block, two distinct categories appear. One for interconnecting company’s sites together and the other
one to allow the company’s sites going to the Internet. The “VPN sites” block contains information such as the pack QoS distribution. No pack QoS exists for the Internet services, so no detail is given.

![Figure 6: Service viewpoint](image)

This viewpoint is useful to give the trends about what packs are sold the most. For instance, if a new pack replaces another one, it shows correlation between the decreasing of the old one and the increasing of the new one.

**Technology viewpoint**

The technology viewpoint focuses on one physical interface and details three points namely information regarding the interface (number of sub-interfaces, number of errors, etc…), the repartition of the bandwidth and the repartition of the services. Most time, the physical interface does not have a bandwidth parameter and does not implement services. Rather, it is the sub-interfaces attached to this physical device, which hold those capabilities.

Figure 7 shows the logical distribution of the bandwidth. The services implemented on the sub-interfaces are also represented.
Notice that the “Access points” block refers to the service viewpoint explained in the former part.

**Client viewpoint**

The client viewpoint shows every single interface or sub-interface configured for a client. For instance, it is important to know when a client resigned and what devices need to be deprovisioned (i.e. resources are no longer set up for this client and are reallocated to someone else). It also shows if one client is provisioned on only one or two PE routers. It means that if one of these routers goes down the whole client network goes down, which may cause serious trouble to the client’s business.

**Regular information about router and interface**

This category gathers information that cannot be associated with any of the previous categories. For example, counting the number of clients, the number of VRF, etc… is interesting, these data
are relevant for a router whereas for an interface the number of sub-interfaces, the number of sub-interface down, etc… are more appropriate.

**History**

There is also a possibility to show a graphical view of the data. It is useful to see in one blink of an eye what is the most sold technology on which pack and so on. Graphical representations are available only for the pack QoS repartition and the bandwidth repartition.

All these different viewpoints combined with the users and the operational departments needs, bring ideas about how the dashboards might be organised. In order to get these data relevant for the different users, several dashboards have been made. Here is the solution provided:

- one dashboard gives a global overview of the router and the pack QoS repartition. It is a combination between the regular information about a router and the service viewpoint.

- one dashboard gives a global overview of the bandwidth repartition per type of services. This is a melting pot of the technology viewpoint and the service viewpoint. The first one details the second one.

- one dashboard gives an overview about the bandwidth repartition over the different interface. It shows trends for bandwidth and pack QoS. This is a combination between the regular information of an interface and the bandwidth repartition of it sub-interfaces (i.e. bandwidth point of view).

- one dashboard gives a list of router interfaces on which a client is provisioned. This is simply the client viewpoint.

- one dashboard shows the distribution of the different services. This is simply the service viewpoint.

As data are stored into the database, a history of the evolution of the data is kept.
4 SARIM: the tool

The former chapter presented the logical representation of the NEs, this chapter emphasises on my own work. The project runs under the acronym SARIM.

This chapter details the way the tool has been implemented. It describes the technologies used and finishes by showing screenshots of the tool.

4.1 Implementation

When realising this tool, it was decided to use standardised formats whenever possible. The code has also been written with flexibility in mind.

First, the database is accessed via some PHP\(^8\) pages. PHP is an HTML-embedded scripting language used to create dynamic pages. The way PHP code opens the socket to talk with the database is independent from the database itself. In order to achieve that, the PHP pages call a set of special classes (e.g. PEAR\(^9\) modules). Changing the database without rewriting the whole queries is easy to do. For instance, if the database needs to get more reliable and powerful, one parameter has to be changed in only one file whereas queries remain the same.

The second point is that the text has been separated, when it was possible, from the presentation. It simply means that if the loading of the style sheet fails, the contents are still available and readable. Of course, the look and feel are not the same but the page still has some consistency.

\(^8\) PHP stands for Personal Home Page, the web page is located at http://www.php.net
\(^9\) PEAR’s homepage http://pear.php.net
Finally, the PHP pages generate fully compliant eXtensible HyperText Mark-up Language (a.k.a XHTML), this allows portability on different devices. This task is complex because pages content is generated dynamically; the challenge resides in the fact that every time a page is generated, it must be XHTML compliant.

Why using XHTML instead of HTML?

The reasons are simple. First, XHTML is not different from HTML. XHTML is basically HTML 4.01 rewritten with XML 1.0 syntax in mind. XHTML is a recommendation chartered by the W3 Consortium in [16]. XHTML is the successor of HTML, so HTML is going to be progressively replaced by XHTML. Finally, XHTML is stricter, more modular and cleaner.

4.2 How pages are generated?

The site presents organised data in preformatted dashboards. They are generated on demand with the powerful combination of the hypertext pre-processor PHP and the MySQL\(^\text{10}\) database. SARIM retrieves data from the database then constructs XHTML-friendly pages. Finally, a Cascading Style Sheets (see next section for further details) file is loaded and applied to obtain the appropriate layout.

In order to create the dashboards, templates (e.g. pre-defined PHP pages) are designed. These templates contain both the layout of the data and the layout of the presentation. It means that the XHTML tags that define the layout of the data are stored into the PHP template page. This is the static part of the page. Between the different tags and sometimes the tags themselves, there might be PHP variables. This is the dynamic part of the page. These variables are filled out when the page is requested by the user. Data are usually retrieved from the database. It is also possible to give the XHTML tags some attributes that hold the styles, which give the look and feel of the site by controlling the layouts of the presentation.

\(^{10}\) MySQL’s homepage http://www.mysql.com
Here the web-site is not a fully-template based one. There is no template engine, which parses the template page and dumps a XHTML page. This task was not the aim of this thesis. SARIM uses pre-defined pages and fills out variables with data. Nevertheless, the amount of data is variable, this is why the pages are called templates.

As dashboards are designed for specific purposes, every dashboard has its own template. More precisely, all the pages in the web site have their own template.

### 4.3 Using Cascading Style Sheets techniques

The classical way of building a web site, is to use the tag attributes to give instructions for the presentation layout. However, as this tool is created with standards and flexibility in mind, it was decided to separate data from content using Cascading Style Sheets.

#### 4.3.1 What are Cascading Style Sheets?

Cascading Style Sheets (CSS) are a recommendation chartered by the W3 Consortium in [17].

Some years ago, CSS was born with the idea to bring independence between data layout and data. The concept of CSS is simple. HTML or XHTML are used for their original purpose, which is holding the content of web pages whereas a specific file contains all the style attributes such as text colour, background colour. The link, between these two, is made with the help of the XHTML tags. The association between the tag and the style is made directly through the tag name or by using an additional attribute (e.g. id attribute or class attribute). Thus, the HTML or XHTML code is written and the special attribute is added. It refers to a section into the CSS file. The section is a block, which gathers a set of attributes (e.g. position and look and fell). This combination gives the pages’ layout and look and feel.
Figure 11 is an example of a CSS file. Bold lines are CSS selector and between the ‘{’ and ‘}’ characters there are the style attributes.

With few effective-positioning parameters combined with the blocks’ layout, CSS is very powerful. This gathers modularity and flexibility; contents remain the same and only layouts change. HTML-based pages remain simple, no “dirty” attributes are added to the tags and only one file holds the styles. To reflect a new layout style, only one file needs to be changed.

The web site design is based on the technique of using blocks. I am going to explain and demonstrate this concept in the section 3.6.3. Because of this design, the site will have consistency even if the CSS file fails to load.

4.3.2 Why using Cascading Style Sheets?

Using CSS allows to have two separated files (i.e. one HTML plus one CSS) and each has a specific and precise aim.

The concept of CSS is simple. Moreover, for people who are used to write pages in HTML, CSS is not very different from what they used to do. Nonetheless, CSS is not widespread for one reason. To understand this reason, let me remind about some facts. At the beginning of the Internet, more precisely of the web sites, only simple texts were displayed. As the technology evolved, contents became more complex and richer. The <TABLE> tag, originally designed to present data in a tabular mode, was quickly adopted to create the template of the site.

The trick consists of removing the border attribute, just by reducing its size to zero. Thus, the border becomes invisible. So the tag turns out to be powerful and allows creating whatever layout is needed. On the other side, a web site exclusively designed with <TABLE> tags is hardly maintainable. It is very difficult to modify it without adding several radical changes into the web pages. This is due to nested tables. All layout changes need to rewrite most of the code.
Finally, one last argument not to use the `<TABLE>` tag is that to compute a page is time-consuming. Again, this is due to the nested tables.

People are still using the `<TABLE>` tag because they are used to it. Nowadays, more and more web sites start using blocks combined with CSS instead.

CSS has not only positive sides, it also has negative ones such as a poor support of this standard. Until recently, browsers were not fully compliant with CSS level 2. Most of the current browsers support CSS level 1 (from Netscape 4.x, Internet Explorer 5.x). Another point has to be noticed browsers do not implement CSS in the same way. So CSS files become dirty because they need to consider browser capabilities. This is not a good thing.

References [19] and [20] are recommended reading to understand and encourage the use of CSS instead of `<TABLE>` tags.

### 4.3.3 A Cascading Style Sheets sample

*Note: when referring to XHTML, HTML can be used as well, this is done in order to not overwhelm the explanations.*

Before getting into the coding part, a distinction has to be made between the layout of the page and the layout of the data. These are two different things. The layout of the page is the way contents are presented. For instance, where is this element positioned? How is it shown? Strong, italic, underlined whereas the layout of the data is the order in which contents appear. This section discusses both layouts.

I already mentioned the main idea, in order to create a web page with CSS, its blocks have to be created. A block is part of the XHTML code that gathers under a unique category a number of
instructions, which have some common points. For instance, the header of a page is considered as one block, the navigation menu is considered as another block and so on. The piece of code is put between the <DIV> and </DIV> tags. These tags are regular XHTML tags.

Here is a sample of one of the generated XHTML pages.

```xml
<div id="navigation">
  <div class="menu0">
    <div class="menuTitle0">
      <a href="/SARIMv2/index.php?viewCategory&amp;c=0">Tableaux de Bord</a>
    </div>
    <div class="menuItem0">
      <ul>
        <li><a href="/SARIMv2/index.php?p=interface">Interface</a></li>
        <li><a href="/SARIMv2/index.php?p=client">Gestion des clients</a></li>
      </ul>
    </div>
  </div>
</div>
```

**Figure 9: XHTML file sample**

The <DIV> tag with the id="navigation" attribute (in bold in Figure 9) is the top-level container. It just contains all the elements related to the web site navigation. Then, a second block appears with the class="menu0" attribute. It contains all the elements connected to the first menu. It is divided into two sub-blocks. One with the class="menuTitle0" which is the title of the category and the other one with the class="menuItem0" which lists the sub-items.

Now the data layout is chosen. If we ask a web browser to render those contents, nothing exciting will appear as illustrated below. The rendering is done under Microsoft Internet Explorer 6.0 (IE 6.0).
In order to give the contents a better layout, the following CSS file is applied.

```css
#navigation {
    float: left;
    width: 140px;
    margin-bottom: 5px;
    border: 1px solid #487725;
}

.menu0 {
    width: 100%;
    margin: 0px;
    font-weight: bold;
    font-size: 14px;
    background-color: #b9e700;
}

.menuTitle0 a {
    color: #487725;
}

.menuItem0, .menuItem0 a {
    color: #ffffff;
}
```

Figure 11: CSS file sample
This file only contains instructions regarding the text colour and the background colour. The next screenshot illustrates the rendering under IE 6.0 browser.

![Image](image.png)

**Figure 12: Block elements rendered with CSS file**

Of course, the crucial point is to design the fraction but it is also to apply positioning styles (i.e. tell where the blocks have to be set). This second point is done with the set of instructions such as float, top, left, right, bottom, margin, etc… Each block combines one or several of these in order to create the most appropriate layout. In the previous example, the layout is achieved with the instruction “float: left“ which indicates that the contents are floated to the left side.

The aim of the project is not to create several layouts for NEs but to create one representation. This is why only one CSS file has been created. The CSS Zen Garden web site, located at http://www.csszengarden.com/, shows how to create different layouts with a simple content. You simply have to select a CSS theme and look what can be achieved with a simple XHTML page combined with a well design CSS file.
4.4 Screenshots of SARIM

This section illustrates the purposes that have been discussed in the former chapters and finally gives the representation of the NEs. As I work for a French company, the language used is French, nevertheless, it should not be difficult to understand the contents and seeing the data organisation should not be a problem.

Note: For confidential reason, parts of the screenshot are blurred and others are changed. Data are given as example and can not be considered as representing the network image.

Screenshots are rendered under IE 6.0 with the CSS file applied. Under other browsers, the rendering can be different.

The first screenshot (c.f. Figure 13) is a dashboard that presents information about a router. As it is described in section 3.3, the dashboard is divided in two parts. On one side, there is the service viewpoint with two sub-categories. On the other side, there are regular information regarding the routers such as the number of client, the number of errors and the number of VRFs.

Figure 13: Services distribution and router’s details

Figure 14 presents the interface viewpoint. It shows two ways of presenting data. First, data are presented in a synthetic way (e.g. ATM0/0/0) or in a detailed way (e.g. ATM3/1/0). The first line is the summary and presents the following information: the interface’s name, the description, the number of sub-interfaces, the number of sub-interfaces down, the number of errors and the status of the interface.
The second line shows in addition to the previous information, the distribution of the bandwidth (i.e. bandwidth viewpoint), the distributions of the VPNs and the Internet services (i.e. service viewpoint).

Figure 14: Interface viewpoint

Figure 15 presents the detailed summary of the services distribution. Two levels of details exist and both services can be detailed. As for example, on the left side, the summary version is shown while on the right side the detailed version is presented.

Figure 15: Services distribution
Figure 16 shows the information regarding one client and presents one line per device containing the following information: the hostname, the interface’s name, the name of the sub-interface, the status, the PVC, the bandwidth, and the IP address.

![Table Image]

Figure 16: Client viewpoint

Finally, Figure 17 shows the services distribution by pack QoS and by technology. This is the more detailed version regarding services that can be obtained.

![Table Image]

Figure 17: Services detailed
5 Conclusion

This chapter gives conclusions about the work that has been realised during this thesis. Then it gives examples of the tool’s uses. Finally, it adds some remarks in order to improve the tools.

5.1 Monitoring Network Elements is possible

Let me remind you about the three study questions that have lead this thesis paper:

- What elements can be monitored within the network of a service provider?
- Is it possible to retrieve these data? If so, how to get them?
- How to combine them so that they become useful?

It is obvious that the two first questions are really easy to answer and the solutions are simple. Yes, there is a number of Network Elements within the network of a SP that can be monitored. And yes, it is possible to retrieve monitoring data and to store them.

Let me add some complementary information about the last point. The solution, I have found, works well in the sense that it successfully retrieves the data. It simply parses the configuration file and retrieves the required information. One negative point is that the script is quite predictive. I mean it has not been programmed to recognise a new technology. Indeed, this new technology is not taken into account until the full chain (i.e. script + gathering process + dashboard) has been updated.

Another negative point is that some information are not relevant using the configuration file as a source of data. For example, monitoring the status of an interface is not relevant with this way of doing it. Consider that the script is launched every Sunday. If the client’s interface is down
because the client has switched off its router for the week-end, there is absolutely no way of finding if the client is still a customer or just disconnected because he has some problems.

Regarding the last question representing the monitoring data in a relevant way is also possible. But and this is an important “but” it also has its limitations because everything is pre-programmed and if data do not fit in the rules, it will not appear on the web page.

I did not succeed to get an universal representation of the data. Rather whatever the data is, there is a way to represent it but depending on what is needed. One representation does not fit everyone. Nevertheless, the tool is flexible enough to tolerate some extra-features and this is a positive point.

5.2 Using SARIM

SARIM is designed for counting elements within the SP’s network and gives a representation of them, the SP’s employees use those representation for several purposes. The following gives examples.

Employees from the marketing department make statistics about the network evolution: grow forecasts, client number, etc…. For instance, one goal could be “before the end of 2005, the company will have 5000 clients and 200000 sites”. There are two ways to know if this is true. First, to use the repository that contains everything that has been sold. Finding this information is not easy because the repository has not been designed for gathering data. The second choice is to use SARIM, which gives a snapshot of the network and provides that information. SARIM has history functionality. Hence, getting the life cycle of pack QoS, correlation between the launching of a new pack and the declining of an old one are simple queries that SARIM can answer.
Employees from the supervision department use SARIM in two major ways. Retrieving information from the network is the main use. For instance, how many interfaces are set up on this router? This brings directly to the second use, which is linked to “capacity planning”. Routers have limits such as computing ones (too many sites, etc…), physical ones (no more card can be added, etc…). In order to know when these limits will be reached, routers have to be monitored. Constructors provide recommendations for their devices so SPs have to compare those given recommendations with the filling rate of the router. With the help of the history function, limit reaching will be quickly seen and provisioning of new devices is planned.

The last use of SARIM is the error detection. Hence, the provisioning team must exploit this source of information in order to correct mistakes.

5.3 Future work

At the beginning of the work, one aim was to create a kind of guideline to represent NEs within the OSS department but quickly the research I made and the conclusion show two things. First, NEs are not so numerous from the OSS point of view. It means that considering the scope of the OSS, there are only a few items available, which have been treated here. Finally, data and their representation are not separable. As data must be gathered following a number of common concerns, it is hard not to link data and its representation.

Now, from a technical point of view, the method I used to represent these elements is functioning. XHTML and CSS are a powerful combination. XHTML assures a nice-coded portable format while CSS handles the layout. Almost all browsers support CSS level 1 and most of them have a good implementation of CSS level 2. The way XHTML pages are coded (i.e. using blocks) ensures that even if the CSS fails to load the contents are visible and clear.
SARIM is not one hundred percent efficient, scripts have to be added to allow it to start automatically in case of crashing. It also has to be moved to the Network Operation Center in order to be fully supervised. After these operations, it would be ready to be fully used.
Bibliography

This section gathers all the references used to write this document


http://www.itu.int/ITU-T/

http://www.ietf.org/rfc/rfc3031.txt

http://www.ietf.org/rfc/rfc2547.txt

http://www.ietf.org/rfc/rfc2598.txt

http://www.ietf.org/rfc/rfc3247.txt

http://www.ietf.org/rfc/rfc2597.txt


http://www.iec.org/online/tutorials/ems/


Appendix 1: Router configuration file

This is a sample of the configuration file extracted from what the company uses. It belongs to a router series that is used as PE router.

Here is a interface example.

```
interface FastEthernet0/1/0
no ip address
no ip redirects
speed 100
full-duplex
no cdp enable
```

Here is a sub-interface attached to the interface describe above.

```
interface FastEthernet0/1.105
description FastEthernet0/1.105 isl vlan id=105 : Provisioned By VPNSC: Service Request Id# = 4990
bandwidth 4096
capsulation isl 105
ip vrf forwarding V1607:XXXXVPN1
ip vrf sitemap SOO-XXX-8
ip address 172.30.196.5 255.255.255.252
no ip redirects
no ip proxy-arp
service-policy output SHAPER-4096-FastEthernet0/1.105-XXXX
no cdp enable
```

Here is a QoS configuration

```
policy-map SHAPER-4096-FastEthernet0/1.105-XXXX
class class-default
shape average 3856000 15424 15424
service-policy P-DM-4096-0-2048-1024-550-XXXX
policy-map GEND3B-M
class C2
bandwidth percent 30
random-detect dscp-based
random-detect exponential-weighting-constant 10
random-detect dscp 16 32 64 9
random-detect dscp 32 64 192 9
queue-limit 256
class C2-B
bandwidth percent 18
random-detect dscp-based
random-detect exponential-weighting-constant 10
random-detect dscp 10 64 192 9
random-detect dscp 12 32 64 9
queue-limit 256
```
class C2-C
bandwidth percent 15
random-detect dscp-based
random-detect exponential-weighting-constant 10
random-detect dscp 18 64 192 9
random-detect dscp 20 32 64 9
queue-limit 256
class C3
bandwidth percent 12
random-detect dscp-based
random-detect exponential-weighting-constant 5
random-detect dscp 0 64 192 7
queue-limit 256
class class-default
queue-limit 256
Appendix 2: CVS file sample

Below, there are some rows extracted from the original CSV file. For a better comprehension, headers are added (they appear in bold, here). For confidential purpose, the following fields VRF, description, client’s name, rt_import, rt_export and description can not appear.

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Interface</th>
<th>Sub-iface</th>
<th>Iface type</th>
<th>Status</th>
<th>PVC</th>
<th>Bandwidth</th>
<th>Pack QoS</th>
<th>Technology</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIL1MC7501</td>
<td>Hssi5/0/0</td>
<td></td>
<td>PHYSICAL</td>
<td>UP</td>
<td>N/A</td>
<td>44212</td>
<td>NONE</td>
<td>FRAME-RELAY</td>
</tr>
<tr>
<td>LIL1MC7501</td>
<td>Hssi5/0/0</td>
<td>2</td>
<td>LOGICAL</td>
<td>UP</td>
<td>824</td>
<td>1920</td>
<td>NONE</td>
<td>FRAME-RELAY</td>
</tr>
<tr>
<td>LIL1MC7501</td>
<td>Hssi5/0/0</td>
<td>3</td>
<td>LOGICAL</td>
<td>UP</td>
<td>825</td>
<td>1792</td>
<td>PACK-STANDARD</td>
<td>FRAME-RELAY</td>
</tr>
<tr>
<td>LIL1MC7501</td>
<td>Hssi5/0/0</td>
<td>4</td>
<td>LOGICAL</td>
<td>DOWN</td>
<td>826</td>
<td>1920</td>
<td>PACK-VOICE</td>
<td>FRAME-RELAY</td>
</tr>
<tr>
<td>LDF1MC7506</td>
<td>FastEthernet2/0/1</td>
<td></td>
<td>PHYSICAL</td>
<td>UP</td>
<td>N/A</td>
<td></td>
<td>PACK-MULTIDATA</td>
<td>ETHERNET</td>
</tr>
<tr>
<td>LDF1MC7506</td>
<td>FastEthernet2/0/1</td>
<td>1</td>
<td>LOGICAL</td>
<td>UP</td>
<td>1</td>
<td></td>
<td>PACK-STANDARD-PLUS-L</td>
<td>ETHERNET</td>
</tr>
<tr>
<td>LDF1MC7506</td>
<td>FastEthernet2/0/1</td>
<td>109</td>
<td>LOGICAL</td>
<td>UP</td>
<td>109</td>
<td>4096</td>
<td>PACK-DATA</td>
<td>ETHERNET</td>
</tr>
<tr>
<td>VAL1MC7501</td>
<td>Serial5/1/1</td>
<td></td>
<td>PHYSICAL</td>
<td>UP</td>
<td>N/A</td>
<td></td>
<td>NONE</td>
<td>PPP</td>
</tr>
<tr>
<td>VAL1MC7501</td>
<td>Serial5/1/1</td>
<td>2</td>
<td>LOGICAL</td>
<td>UP</td>
<td>400</td>
<td>1920</td>
<td>PACK-DATA</td>
<td>PPP</td>
</tr>
<tr>
<td>VAL1MC7501</td>
<td>Serial5/1/1</td>
<td>34</td>
<td>LOGICAL</td>
<td>UP</td>
<td>401</td>
<td>256</td>
<td>PACK-DATA</td>
<td>PPP</td>
</tr>
<tr>
<td>VAL1MC7501</td>
<td>ATM3/1/0</td>
<td></td>
<td>PHYSICAL</td>
<td>UP</td>
<td>N/A</td>
<td></td>
<td>AAL5SNAP</td>
<td></td>
</tr>
<tr>
<td>VAL1MC7501</td>
<td>ATM3/1/0</td>
<td>1</td>
<td>LOGICAL</td>
<td>UP</td>
<td>N/A</td>
<td>607</td>
<td>PACK-VOICE</td>
<td>AAL5SNAP</td>
</tr>
<tr>
<td>VAL1MC7501</td>
<td>ATM3/1/0</td>
<td>4</td>
<td>LOGICAL</td>
<td>UP</td>
<td>(1/247)</td>
<td>607</td>
<td>PACK-STANDARD-XS</td>
<td>AAL5SNAP</td>
</tr>
</tbody>
</table>