AALBORG UNIVERSITY
STUDENT REPORT

IMPROVEMENT OF CELLULAR PERFORMANCE BY OFFLOADING MOBILE SUBSCRIBERS TO LOCAL xDSL CONNECTIONS

NINTH & TENTH SEMESTER

Master's Thesis

NETWORKS & DISTRIBUTED SYSTEMS

Authors:
Daniel Mandrup
Piotr Remlein
Jesper Graversgaard Thomsen

Supervisors:
Tatiana Kozlova Madsen
Martin Fejrskov Andersen

June 1, 2016

Denmark

AALBORG UNIVERSITET

Ninth & Tenth Semester
at School of Engineering and Science

Networks & Distributed Systems
Frederik Bajersvej 7
Phone 96 35 97 31
Fax 98 13 63 93
http://www.sict.aau.dk/

Title:

Improvement of cellular performance by offloading mobile subscribers to local xDSL connections

Theme:

Performance analysis and network planning

Authors:

Piotr Remlein
Jesper Graversgaard Thomsen
Daniel Mandrup

Project period:

P9, Autumn 2015
P10, Spring 2016

Supervisors:

AAU: Tatiana Kozlova Madsen
Telenor: Martin Fejrskov Andersen

Number of copies: 6

Number of pages: 197 (252 with appendix)

Report Finished: 01-06-2016

Synopsis:

The demand for mobile data is ever increasing and is causing huge stress on cellular networks. Upgrading the capacity of the cellular network is an expensive and time consuming process, and the demand does not seem to slow down. The idea is to offload mobile cellular subscribers to the wired xDSL connections. This project attempts to design and implement an innovative and cheap solution by utilising the spare capacity of the xDSL connections. Doing this introduces four important issues: Network Integrity, Quality of Service, Authentication, and User Identification.

Network Integrity

Network integrity is important to ensure the privacy of both the xDSL subscriber and the mobile subscribers. This is achieved by creating a separated WiFi instance for the mobile subscribers and adding a series of strict firewall rules. Various vulnerability scans have been performed to test the implementation, and they show no signs of security issues.

Quality of Service

QoS is a necessity to ensure that the xDSL subscriber's paid service is not degraded by the implementation. Several QoS mechanisms are implemented, and traffic identification for each network is applied through port translation. This enables Telenor to apply QoS on their side of the network for downstream traffic. Measurement results shows that upload QoS can not always be ensured in cases where the wireless medium is congested. Download QoS should in theory be ensured when QoS is applied on Telenors side, as long as the bottleneck is on the xDSL link and not the wireless medium.

Authentication and User Identification

Only Telenor mobile subscribers must be able to connect to the new WiFi instance, and the data of each individual user is required to be logged according to the Danish laws of data retention. A RADIUS server is implemented to authenticate and log subscriber information. To log the correct data to the specific user, traffic identification is applied through port translation. Various tests have been made to make a proof of concept.

Signatures:

Piotr Remlein

Jesper Graversgaard Thomsen

Daniel Mandrup

PREFACE

This master's thesis has been made as a fulfillment of the master programme 'Networks and Distributed Systems', at Aalborg University. The master's thesis is a product of the efforts of two semesters: Autumn 2015 and Spring 2016.

Acknowledgements

We would like to thank our head supervisor Tatiana Kozlova Madsen, Aalborg University, for a lot of valuable advice as well as continued patience and support during the preparation of the thesis.

Thanks also to Martin Fejrskov Andersen, co-supervisor from Telenor Danmark, for the extensive technical expertise as well as patience and understanding for our endless questions and long meetings.

We are sincerely thankful for the time Tatiana, Martin, and our fellow students have devoted to us, and for their contribution in shaping a friendly and pleasant atmosphere that is invaluable for the scientific work. All their suggestions have allowed us to better understand the scientific topics and to create a better thesis. Their directions has been very important for the formation of our work.

Conventions used in this project

The first time an acronym is used the whole word/words is written followed by the abbreviation, for instance, Round Trip Time (RTT). Thereafter the abbreviation will be solely used. A list of all the abbreviations can be found in the beginning of the report.

Throughout the report the references are based on the Harvard method with the author's last name in square brackets [ ]. The references will be placed at the locations where they are referred to and a gathered list of the references can be found in the Bibliography section in the last pages of the report where they are sorted according to the author's last name.

Figures, tables and equations are numbered according to which chapter they are located in and the order in which they appear in the specific chapter e.g. the first table's number in chapter four is 4.1 and the second is 4.2 etc.

Relevant configuration guides, configuration commands, and code developed for this project as well as other files can be found on the group web page: `http://kom.aau.dk/group/16gr1022/`.

Several types of listings appear in the project that can be distinguished and should be understood as follows:

- Listings with a white background are used to show configuration commands issued to the MediaAccess Gateway.
- Listings with a gray background are used to show the settings of the MediaAccess Gateway.

  All types of MediaAccess Gateway listings will use the following color scheme:

  - *Menu* in which the configuration is.
  - *Command Line Interface (CLI) command.*
  - *message from the MediaAccess Gateway* when a command is executed.

- Listings with light blue background are used to show a result of *nmap*[1] scans.
- Listings with light red background are used to show RADIUS server configuration settings.
- Listings with a light green background are used to show the content of RADIUS server accounting messages.

○ [...] is used when a part of a listing has been omitted.

○ In *nmap* and RADIUS server listings different colors will occur to emphasize the importance of a given output.

[1] nmap: is a tool to analyse vulnerabilities in networks. This could for example be by looking for open TCP ports.

ACRONYMS

Acronyms

AAA Authentication, Authorization, and Accounting

AC Access Category

ACS Auto Configuration Server

ADSL Asymmetric Digital Subscriber Line

AES Advanced Encryption Standard

AP Access Point

ARP Address Resolution Protocol

BNG Broadband Network Gateway

BRAS Broadband Remote Access Server

BSSID Basic Service Set Identifier

CAGR Compound Annual Growth Rate

CLI Command Line Interface

CPE Customer Premises Equipment

CTO Cellular Telecommunication Operator

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

DSL Digital Subscriber Line

DSLAM Digital Subscriber Line Access Multiplexer

EAP Extensible Authentication Protocol

EDCA Enhanced Distributed Channel Access

FAP Fair Access Policy

FTP File Transfer Protocol

ICMP Internet Control Message Protocol

IP Internet Protocol

ISP Internet Service Provider

LAN Local Area Network

MAC Media Access Control

MAG MediaAccess Gateway

NAPT Network Address and Port Translation

NFS Network File Server

NAS Network Access Server

NAT Network Address Translation

NetBIOS Network Basic Input Output System

OBC Onboard Controller

OS Operating System

PoP Point of Presence

PPP Point-to-Point

PPPoA Point-to-Point over ATM

PPPoE Point-to-Point over Ethernet

PSK Pre-Shared Key

QoS Quality of Service

RTT Round Trip Time

SDSL Symmetric Digital Subscriber Line

SSID Service Set Identifier

TCP Transmission Control Protocol

TKIP Temporal Key Integrity Protocol

ToS Type of Service

UDP User Datagram Protocol

VDSL Very-high-bit-rate Digital Subscriber Line

VID VLAN Identifier

VLAN Virtual Local Area Network

VoIP Voice over IP

VSA Vendor-Specific Attributes

WAN Wide Area Network

WEP Wired Equivalent Privacy

WLAN Wireless Local Area Network

WMM Wi-Fi Multimedia

WPA Wi-Fi Protected Access

WPA2 Wi-Fi Protected Access version 2

CONTENTS

CHAPTER

ONE

INTRODUCTION

The demand for mobile data usage is ever increasing and is expected to be rapidly increasing in the coming years [Cisco [2016]] [Ericson [2015a]] [Ericson [2015b]]. The increase is mainly due to mobile users who wish to be *connected at all times* when the applications and services they are using are gaining more ground. The trend is that some of these applications and services are relying heavily on high speed connections. A few examples to mention are Facebook, that requires an *always-on* service, or on-demand-video streaming services like Netflix, YouTube, and HBO GO. The Cellular Telecommunication Operators (CTOs) are struggling to keep up with the pace, that are currently threatening to overload the capacity of their cellular networks.

A summary made by Cisco [Cisco [2016]] shows that, during 2015, cellular traffic grew 52 percent during 2015 in Western Europe and 71 percent in Central and Eastern Europe, whereas Ericson [Ericson [2015a]] suggests that the growth was 45 percent for all of Europe[1]. The summaries made by Cisco and Ericson both estimate that smart devices[2] represented around half of the total mobile devices being connected in 2015. Both summaries agree that smart devices account for almost all mobile data traffic because a smart device uses a far greater amount of data compared to a non-smart device. On average a smart device generated 14 times more traffic than a non-smart device in 2015 [Cisco [2016]].

It is predicted that not only will mobile data usage increase rapidly in the coming years, but the amount of mobile subscriptions will increase slightly every year as well. This increase in the amount of subscriptions is going on while mobile subscribers are also switching from non-smart devices to smart devices [Cisco [2016]] [Ericson

[1] It is chosen to only show data from Europe as some parts of the world/countries are currently implementing and upgrading cellular networks at very rapid rates.

[2] A smart device is here defined as a mobile device that have advanced multimedia/computing capabilities with a minimum of 3G connectivity.

[2015a]] [Ericson [2015b]]. Figure 1.1 shows the expected distribution of mobile subscriptions between the years 2011 and 2021 for Europe. It shows that the amount of mobile subscriptions will increase slightly while the amount of GSM/EDGE-only subscriptions will fade to almost nothing in 2021. This means that a great deal of mobile devices connected in 2021 will be smart devices and will account for roughly all of the mobile data traffic, making the gap between traffic generated by smart devices versus non-smart devices extremely high.

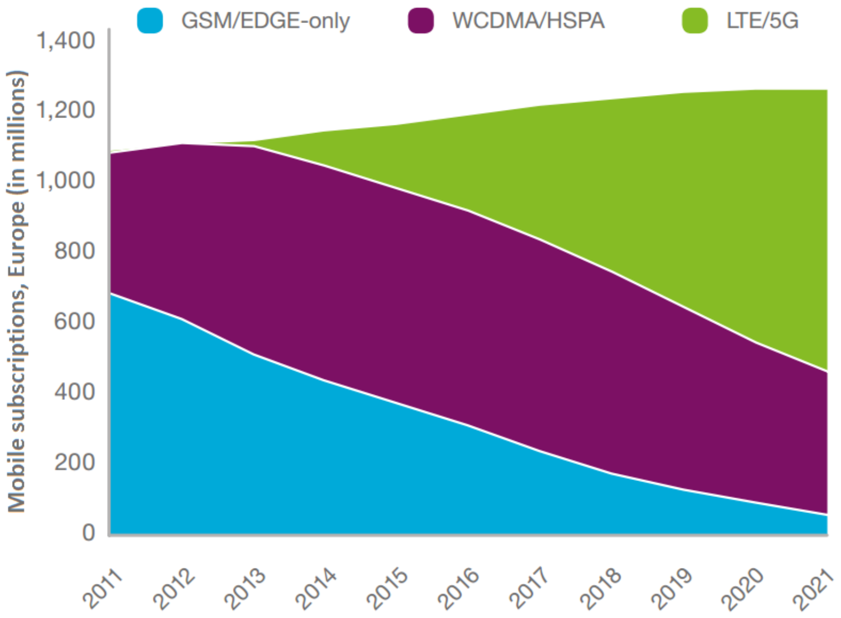


Figure 1.1: The distribution of mobile subscriptions, in millions, in Europe between 2011 and 2021 [Ericson [2015a]].

The growth of both smart devices and non-GSM/EDGE subscriptions contributes to a large growth in traffic flowing through the mobile network. Figure 1.2 illustrates the expected growth in data traffic in the coming years. It is seen that data traffic, in Europe, is expected to grow from a total 1.8 exabytes per month in 2015 to 13 exabytes per month in 2021. One of the reasons for this growth is because of the introduction of new applications and services using large amounts of data, while the development and introduction of new cellular technologies also increases the possible speed for each device. From the given Figures, it shows that the growth is 45 percent Compound Annual Growth Rate (CAGR) between 2015 and 2021 meaning that the increase is greater than Moore's law[3] with a CAGR of 42.41 percent. In effect the amount of mobile traffic will increase faster than the increase of processing power, which leads to various problems.

Some of the challenges that the CTOs are going through is the increasing amount

[3] Moore's Law refers to the doubling of processor, storage, and related computing capability every two years. The CAGR for doubling every two years is 41.42 percent increase per year [Murtagh [2012]].

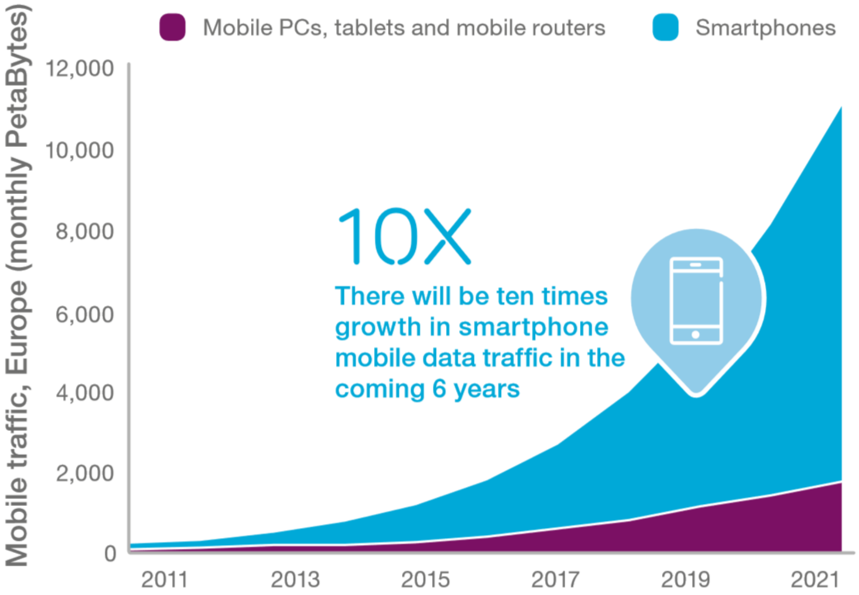


Figure 1.2: The expected growth in monthly mobile data traffic, in Europe, from 2011 to 2021 [Ericson [2015a]].

of cellular users with an increasing demand for greater performance at any time. It means that the CTOs need to invest in the development and introduction of new cellular technologies[4] to gain additional cellular capacity, giving rise to an even greater demand of performance by their mobile subscribers. The trend does not seem to be slowing down with the introduction of new demands like the Internet of Things, an increment in the amount of connections per device, and high resolution video-on-demand streaming will be some of the main contributors to the extreme increase in the amount of capacity required. Moreover the CTOs have limited or bad indoor-coverage due to heavy signal fading ($\approx$ 30-35 dB), mainly due to the green building materials being used. Another limiting factor is Shannon's Limit of maximum capacity based on the signal-to-interference-and-noise-ratio and the modulation/coding scheme being used. Shannon's Limit gives rise to the continuing development of new technologies or the need for additional capacity increment by e.g. doing cell splitting and sectoring.

By doing cell splitting one can reduce the size of a cell, and thereby the transmitting power, of each cell to get higher capacity because the channel reuse distance decreases. By doing cell sectoring one can gain capacity by directing channels into specific sectors of a cell [Sørensen [2016]]. It is very beneficial to do cell splitting and sectoring to increase cellular capacity, but they can be rather short term solutions since the demand does not seem to stabilise anytime soon. Adding new cell-towers is extremely expensive and at some point it becomes unpractical due to cell towers being very close to each other. Another way to gain more capacity is to add femtocells which is a micro cell-tower the size of a router for home/office purposes, connected directly through a

[4] Ericson might be ready to deliver 5G performance by 2020 [Eric Kornum [2016]]

broadband connection. The disadvantage of femtocells is that one needs to be in very close proximity to be connected and can only serve a very small range of users (4-16) [Rouse [2013]] making the use of femtocells very expensive.
Alternatively, when experiencing bad coverage, some private costumers take the problem into their own hands and buy signal repeaters/boosters to gain additional signal strength in remote areas. The issue with signal repeaters is that they might give the user a better signal, but they introduce signal interference resulting in decreased capacity for everyone within its proximity. This is why several countries, including Denmark, have banned them. Failure to do so may result in heavy fines [Version2 [2014]].

During 2015 it was seen, on a global basis, that 50 percent of the total traffic generated by mobile devices were offloaded to fixed local WiFi or femtocell connections [Cisco [2016]][5]. Cisco states that much of this offload was done to private WiFi networks, paid by the broadband subscriber himself, because most data usage goes on at the users home. An idea, to let even more traffic be offloaded to local WiFi connections, is to let mobile subscribers connect to WiFi connections belonging to other Digital Subscriber Line (DSL)[6] broadband subscribers if there is unused capacity on the broadband connection. The idea seems rather long term as routers with WiFi capabilities are to be found in almost every household. Offloading mobile subscribers to local WiFi connections is a cheap way to help solve part of the capacity issues experienced by CTOs while also solving the issues with bad indoor coverage. One of the reasons why it can help solve the capacity issues is because fewer users might be connected to the cellular network and seen from CTO's point of view, it is less expensive to serve costumers through xDSL broadband connections compared to a cellular connection.

The idea of offloading mobile subscribers to local WiFi connections is not new and some companies have already put much research into the area; especially in The US [Tofel [2013]] and Australia [Telstra [2016]]. YouSee, a Danish CTO and Internet Service Provider (ISP), belonging to TDC, introduced the concept in the Customer Premises Equipment (CPE) belonging to YouSee customers in the beginning of March 2014. Already at the end of March 2014 they had to shut down the project because of security issues and users complaining about degraded performance on their xDSL broadband connection [Sandal [2014]] that they pay for! The attempt was a failure from both a commercial, technical and publicity perspective.

The purpose of this project, in cooperation with the CTO and ISP Telenor, is to look into the possibility of implementing a sustainable solution than can offload mobile connections to local WiFi connections. Telenor has around 150.000 xDSL customers, most of which have Telenor-owned xDSL routers with various capabilities. The next chapter will present a system overview and a problem statement, leading to the system requirements.

[5] By offloading it is meant that the user switches his traffic usage from a cellular connection to a WiFi connection [Cisco [2016]].

[6] The term xDSL will be used throughout the report as the *x* indicates a wildcard that indicates any DSL technology e.g. Asymmetric Digital Subscriber Line (ADSL), Symmetric Digital Subscriber Line (SDSL), or Very-high-bit-rate Digital Subscriber Line (VDSL) to mention a few.

CHAPTER

TWO

SYSTEM OVERVIEW

This chapter will provide an overview of how mobile subscribers can be offloaded to local WiFi connections to deal with the issues presented in Chapter 1. The concept will be described shortly and Section 2.1 will define the problem statement that will guide the project. Section 2.2 will show a scenario of how the system will work in order to give the reader an understanding of the system as a whole. Section 2.3 will present the overall system requirements and a delimitation of the project.

The idea is to offload mobile subscribers to local WiFi connections to protect the cellular network from overloading, as well as improving indoor coverage. The project will be done in cooperation with Telenor, that is not only a CTO, but also an ISP, which means that Telenor has xDSL subscribers, each having a CPE providing a wired xDSL connection to the Internet[1]. Since the CPE has WiFi capabilities it can be used, not only by the xDSL subscriber, but anyone who is within range of the signal. The idea is to use the capabilities of the CPE to provide a free WiFi solution for Telenor mobile subscribers. The extra gain is obtained by utilising any spare capacity on the xDSL subscribers connection.

Offloading mobile subscribers to wired connections can help Telenor to manage the ever increasing demand and provide better indoor coverage as a bonus. The solution is not expected to solve the entire demand problem, but will work as a cheap and smart solution to attempt to utilise already available infrastructure. One might think that the task to create a free WiFi solution is relatively simple, as YouSee might have thought (See Chapter 1)[Sandal [2014]], but there are several complications and challenges to be met.

[1] The CPE distributed to Telenor xDSL subscribers is the Technicolor TG788vn v2 modem.

2.1 Problem Statement

The goal is to design and implement a solution that will not, in any way, degrade the performance of the xDSL subscribers connection with regards to Quality of Service (QoS) or infringe the network integrity. The xDSL subscriber is the one paying for the connection, and it is only fair to say that the quality of that connection should not be degraded, thus the connection must be in favor of the xDSL subscriber in every case. From a business perspective, Telenor is in fact required to deliver the service that the xDSL subscriber pays for. The network integrity is ensured by preventing data from becoming lost, garbled or modified without consent. This also means that Free WiFi users will not have access, in any way, to any of the xDSL subscribers devices, network drives and services, or shared folders as well as administration of the CPE itself. Besides protecting the network integrity from the perspective of the xDSL subscriber, it must also be protected in favor of individual users of the Free WiFi solution. This is to ensure that the network integrity, seen from the mobile subscriber, remains intact, independent of the access medium used i.e. form WiFi or cellular. As a business perspective, Telenor has some requirements regarding the solution: One is that only Telenor mobile subscribers are allowed access to the Free WiFi network, no extra software is required by the mobile subscriber, and that the activities of individual mobile subscribers are uniquely identifiable in terms of the data retention policies of Denmark. Each ISP is required by law to log subscriber activities, which is e.g. traffic and phone call information as well as information about geographical location.

The free WiFi solution takes basis in what is known as a hotspot solution, and from now on it will be mentioned as so. The project will derive an extended analysis of the issues and challenges mentioned above to answer the following problem statement:

"How can one design and implement a hotspot solution for Telenor mobile subscribers on the Technicolor TG788vn v2 modem, that guarantees no loss of QoS for the paying xDSL subscriber, ensures network integrity, and follows the laws of data retention?"

Looking into the problem statement it can be separated into four categories:

1. ... ***for Telenor mobile subscribers***...
2. ... ***guarantees no loss of QoS for the paying xDSL subscriber***...
3. ... ***ensures network integrity***...
4. ... ***follows the laws of data retention***...

By analysing the four categories it is possible to see that categories 1 and 4 rely on the same setup seen from a network perspective i.e. user authentication and user identification in the network. Category 2 relies on QoS, while category 3 is network integrity seen from a local perspective i.e. separation of networks and/or users. The four categories identifies three parts that will each be treated throughout the report. To answer the problem statement and to accept the final system it is required that each part is fully endorsed. The three parts are described as follows:

Part I: Network Integrity

Network integrity is important to ensure the privacy of both the xDSL subscriber and mobile subscribers. It is ensured by creating a setup that prevents data from becoming lost, garbled or modified without consent in terms of the xDSL subscriber as well as individual hotspot users. It also means that hotspot users will not, in any way, have access to any of the xDSL subscribers devices, network drives and services, or shared folders as well as administration of the CPE itself.

Part II: Quality of Service

QoS is required such that mobile subscribers are not worsening the performance of the xDSL connection seen from the perspective of the xDSL subscriber, who pays for the connection. The performance of the network must, at any time, be in favor of the xDSL subscriber, meaning that mobile subscribers can only utilise spare capacity.

Part III: Authentication and User Identification

Authentication is a necessity to ensure that only Telenor mobile subscribers are allowed access to the hotspot network. User Identification[2] lies in relation to the Mobile subscribers as well as to protect the xDSL subscriber from wrongful accusations. ISPs are required by law to do data retention such as logging and recording user information and their activities. This means that traffic generated by individual Mobile subscribers must be identifiable from each other and not just as the xDSL connection as a whole.

Each part is independent of one another, and they will be treated as such. The report will be structured into three parts, where the knowledge of one part is irrelevant to the understanding of the other. *It is however recommended that the report is read in the way it is structured.* Each part will have the following structure:

- Analysis
- System Design
- Implementation
- Test
- Conclusion

The following section will give an overview of the purpose of the system in order to later define the requirements for the whole system.

[2] In this project user identification is defined as the ability to intercept traffic and relate that traffic to a specific mobile subscriber.

2.2 System Description

This section will provide the reader with an understanding of the concept behind the hotspot solution as well as definitions of the system as a whole. A few scenarios of how the hotspot solution will work, will be shown in order to give the necessary requirements for a system to be fully functional, as well as accepted in terms of an acceptance test.

Figure 2.1 shows the idea behind the term *offload*. The figure shows how a mobile subscriber is connected to the cellular network for voice calls and text messaging, and connected as a hotspot user to a local wired connection in order to transfer data. As Telenor is both a CTO and ISP, both networks are connected to the same core network at Telenor, which means that it does not matter which access medium is used, seen from the perspective of the mobile subscriber.

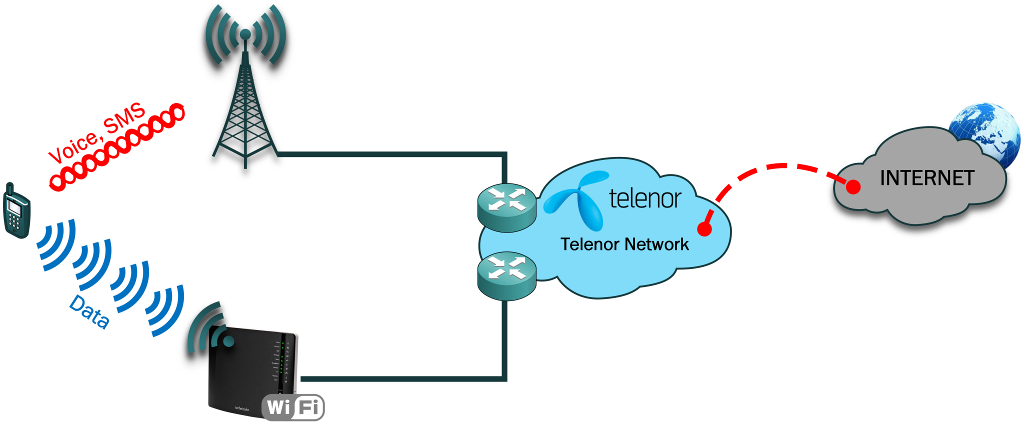


Figure 2.1: The mobile subscriber is still connected to Telenor's core network even though the data is being offloaded.

Figure 2.2 further illustrates the concept of offloading, in terms of a mobile subscriber changing his geographical location. At point **(1)** the mobile subscriber is connected to the cellular network and will have voice calls, text messaging, and data traffic through that connection. When moving, a mobile subscriber might pass by a *hotspot location* as shown in point **(2)**. At this point the mobile subscriber will connect to the hotspot network and become a hotspot user. Voice calls and text messaging traffic will go through the cellular network while data traffic will go through the wired xDSL connection. When the hotspot user/mobile subscriber decides to move away from the *hotspot location* and he will lose connection to the hotspot network as shown in point **(3)**. The mobile subscriber is now connected to the cellular network and voice calls, text messaging, and data traffic will go through that connection.

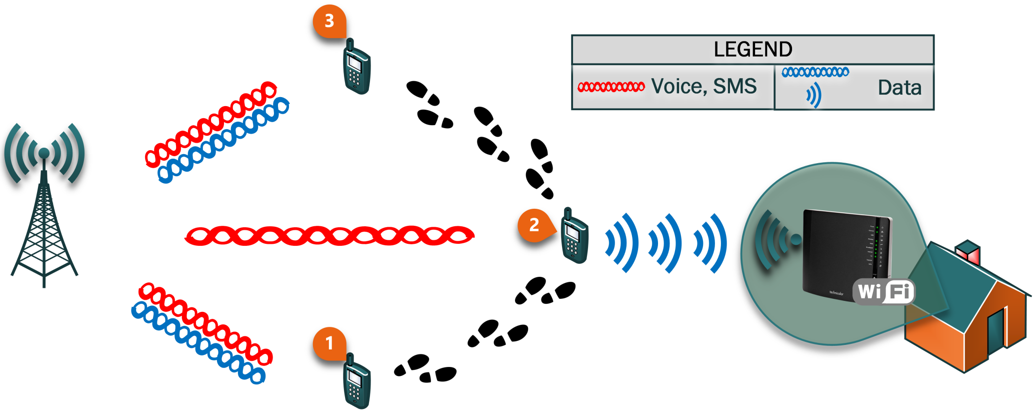


Figure 2.2: Three different scenarios separated by time and geographical location. At points **(1)** and **(3)** the mobile subscriber is connected to the cellular network. At point **(2)** the mobile subscriber is offloading data traffic to the xDSL connection.

2.3 System Requirements

The system requirements takes basis in the problem statement defined in Seciton 2.1 as well as the scenarios just considered:

Telenor mobile subscribers

1. Only Telenor mobile subscribers should be able to use the hotspot network.
2. Traffic generated by individual hotspot users and the xDSL subscriber should be identifiable.

xDSL subscriber integrity

3. Hotspot users can only utilise the spare capacity of the xDSL connection, so that the xDSL subscriber will not detect the slightest degradation of performance.
4. The network integrity must not be compromised for the the xDSL subscriber as well as mobile subscribers.

User friendliness

5. Mobile subscribers do not need to do any WiFi network configuration on their device.
6. The solution does not require any extra software installations.
7. Mobile subscribers should seamless and *automatically*[3] connect to the hotspot network when in close proximity.

[3] A mobile subscriber should only connect if the mobile device has WiFi turned on.

General

8. The hotspot solution will be implemented on the Technicolor TG788vn v2 modem.
9. Every hotspot solution should be identical. *In relation to requirement 8., it should be possible to give each xDSL modem the same configuration without specific changes.*
10. The hotspot solution will only serve data traffic. Voice calls and text messaging will go through the cellular network.

Project Delimitation

As the project considers a very large spectrum of tasks, it is necessary to present a delimitation, presenting what aspects of the solution the project will not solve. The project will thereby not deal with:

Handover procedures: The project will not deal with the handover between cellular networks and the hotspot network. Moreover, it will not deal with the problem of termination of network services when making a handover.

Data logging: The project will provide the necessary tools to enable data logging and recording of user information, but will not deal with an implementation at Telenor's core network.

Only at the CPE: The project will consider all parts of the system, i.e. the network between the CPE and the Internet, but will only do an actual implementation at the CPE, because of restricted availability to Telenor's core network devices.

The concept of offloading mobile subscribers to wired broadband connections through WiFi has been analysed, described, and the requirements have been presented along with a project delimitation. The last index of the project delimitation might give rise to questions as to what Telnor's core network devices are. Appendix A on page 199 aims to give the reader a quick and general understanding of the network considered in this project.

System requirement *7.* announce that *"The hotspot solution will be implemented on the Technicolor TG788vn v2 modem"*. A quick introduction to the features of the Technicolor TG788vn v2 modem is given in Appendix B on page 203. When mentioning specifically that xDSL modem, in the report, it will be referred to as the MediaAccess Gateway (MAG).

The report will, from this point, be split into the three different parts presented in Section 2.1, each trying to solve its own issues. It is possible to read individual parts without prior knowledge of its preceding parts, but it is recommended to read the report in the order it is structured.

Part I

Network Integrity

CHAPTER

THREE

INTRODUCTION & ANALYSIS

The purpose of network integrity is to ensure the safety of everyone, who is going to be involved in the final solution. This includes both the xDSL subscriber, also known as the private user(s), and the mobile subscribers, also known as hotspot users, as safety issues can harm Telenor's public image. As mobile subscribers will gain Internet access through the xDSL subscriber's xDSL modem, it is necessary to ensure the right level of privacy. There are several possible security threats that might occur by allowing a large span of different users to share the same xDSL modem. First of all, the mobile subscriber could obtain an access to the private users devices and network attached drives. That can result in the mobile subscriber modifying, deleting or stealing the private users data e.g. photos, financial information, or private correspondences. The information can be used to gain illegal income and profits by selling the information to third parties, or it can be used to blackmail the owner. Another possibility is that the devices of the private user could be infected by malicious software and used for illegal purposes, without the knowledge of the owner. Of course, all of this can happen in other ways, but as Telenor allows mobile subscribers to connect to the xDSL modem, Telenor has to guarantee that the privacy it not affected.

In order for a hotspot solution to work, there is a need of great separation between the two types of users, which is a private user that pays for the xDSL subscription, and mobile subscribers using the link for free Internet access. When Telenor pushes the solution to the production/live environment, it will for sure be analysed by private entities/companies to test the network integrity. Unexpected findings that will show vulnerabilities in the solution, or even worse, if any of the users will become a victim due to insecure gaps, can have serious consequences for Telenor's public image. It will result in loosing customer trust and thereby decrease the subscription numbers instead of increasing both the number of new customers and good appearance. The task of ensuring network integrity is nontrivial and requires a deep understanding of influenced parts. A very accurate and reliable approach must be taken in order to take into consideration all possible aspects and their influences to the solution. When the solution has been made, various tests must be performed in order to make sure that no mistakes have been made and that the solution acts as expected.

Chapter 3 and Chapter 5 are based on *ConfigGuide_EthernetVLAN.pdf*, *ConfigGuide_Ethernet.pdf* & *ConfigGuide_IPQoS.pdf* found at [Technicolor [2012b]] as well as *CLI REFERENCE GUIDE R10.4 .pdf* [Technicolor [2012a]] unless otherwise specified. However, because of the different CPE model and special software designed purely for Telenor, a large parts of this Chapters have required several experiments and educated guessing in order to effectively learn how different components work and affect each other .

This chapter will focus on detailed analysis and description of the MAG device. In the analysis the following topics will be introduced: Overview of the internal structure of the MAG, introduction to the network interfaces in the MAG, the three network services Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Network Address Translation (NAT), as well as wireless security and authentication.

3.1 MediaAccess Gateway Internal Structure

In order to be able to configure the MAG and to design the best solution for the hotspot network, there is a need for an in depth understanding of the internal structure and processes that occur inside the MAG and its different built-in components, as well as the way these components (bridge, router, access point) interact to each other. To better grasp the concepts during this analysis, a general picture of the basic MAG components and their relationship is presented in figure 3.1 on the facing page.

The different components in Figure 3.1 have the following capabilities:

- **Access point:** The access point allows for wireless connections to the MAG. It has possesses 2x2 MIMO antennas but only one radio with capabilities of creating up to four unique Basic Service Set Identifiers (BSSIDs) (virtual APs), each having a connection to the bridge.
- **Switch/Bridge:** The switch and bridge are two layer 2 devices used for packet switching within the network. The MAG supports the creation of multiple bridge instances, where each bridge can create and maintain multiple VLAN. Compared to the bridge, the switch is very simple as its main function is layer 2 forwarding. However, due to their very different features they require more attention and explanation that can be found in Section 3.2.2.
- **Router:** The router is a layer 3 device with capabilities of forwarding packets between different subnetworks within the MAG and to the Internet. The router

has integrated NAT, DNS, DHCP, and Firewall capabilities, which are presented later in Section 3.3.

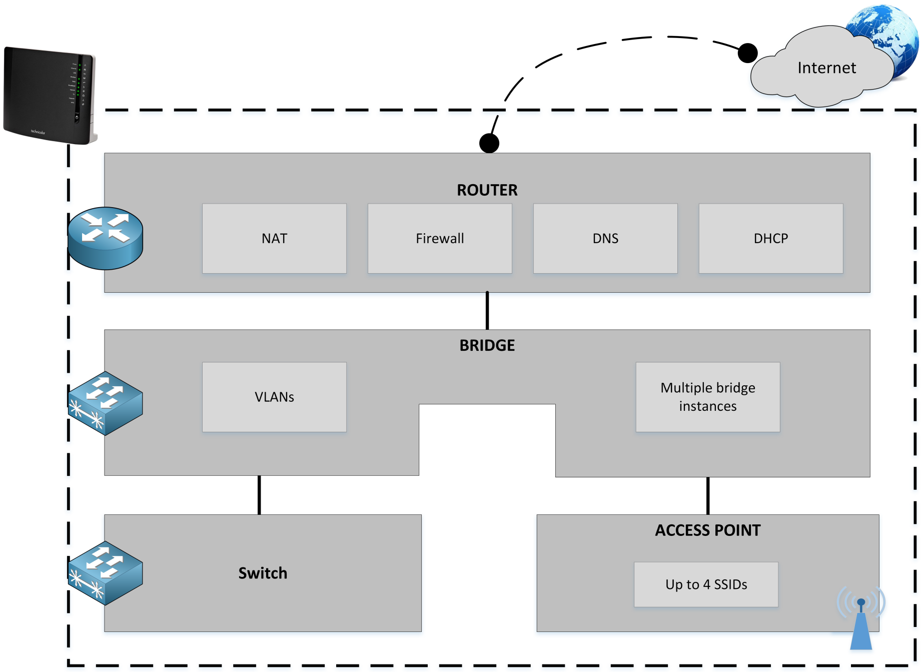


Figure 3.1: General overview of the components in the MAG and their relationships.

3.2 Interface Architecture

In order to enable communication between the components presented in Section 3.1 a set of network interfaces has to be configured. The interfaces enable to connect and disconnect components as well as message passing between them. This section will describe the various types of interfaces in the MAG.

The list of existing interfaces and their connections can be displayed using the Command Line Interface (CLI) command => *interface list*. From the list different interface names, type, state, and their upper layer connection to another interface can be extracted. Depending on the parameters used, either the upper layer or lower layer interface connections can be seen. Based on the list it is thereby possible to figure out the connections of interfaces in the MAG.

It is important for the purpose of the network integrity part to understand how interfaces are connected. To get the default interface configuration of the MAG, Figure 3.2 is created based on the lower layer interface connections, which can be obtained with the CLI command: => *interface list expand=disabled reverse=enabled*, seen in Listing C.1 on page 206 in Appendix C. It should be noted that the three physical ports

FXS1, *FXS2* (physical analog phone ports) and *ndisveth* (usb port used for sharing printers/files) seen in Listing C.1 on page 206 are not shown in Figure 3.2 as their placement is not of particular interest.

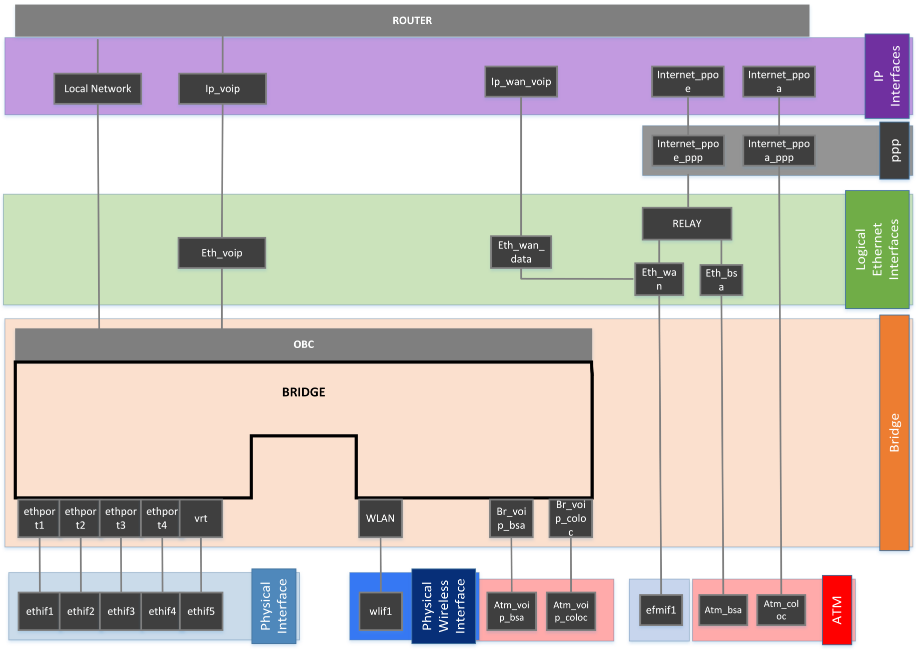


Figure 3.2: Default configuration of interfaces in the MAG.

The interfaces that are important for this project will be presented and described in detail in following Sections. Figure 3.3 has been created in order to easier grasp the overall picture of the particular placement of some of the following key interfaces:

- Physical Ethernet and Physical wireless interfaces e.g. *ethif4* (Section 3.2.1)
- Ethernet bridge ports e.g. *ethport2* (Section 3.2.3)
- Logical Ethernet interfaces e.g. *Eth_voip* (Section 3.2.4)
- IP interfaces e.g *Ip_voip* (Section 3.2.5)

As it is also important to understand the function of the Ethernet Switch and the Ethernet Bridge, Section 3.2.2 that explains their relationship will also be introduced even though it is not an interface as so. Lastly, Section 3.2.6 will combine the key interfaces and give an introduction to the bridge and the router forwarding process.

3.2.1 The Physical Ethernet and Physical Wireless Interfaces

The MAG has both Physical Ethernet interfaces and Physical wireless interfaces as seen in Figures 3.2 and 3.3. The following sections will describe each of the interfaces.

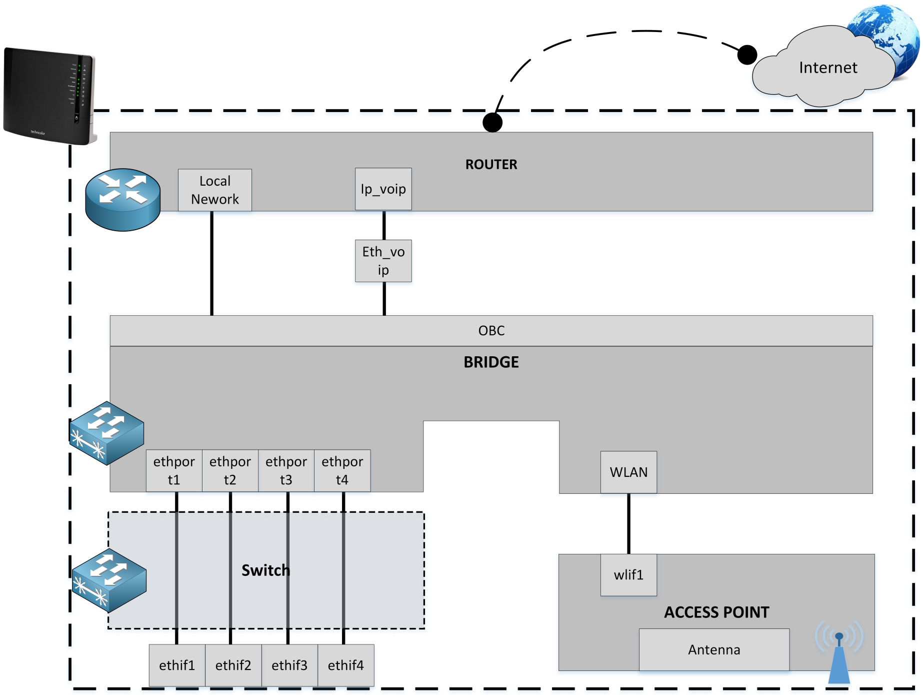


Figure 3.3: Placement of important interfaces in the MAG.

The Physical Ethernet Interfaces

The Physical Ethernet interfaces allow for wireline Ethernet using one of the (default) ports: **ethif1**, **ethif2**, **ethif3**, **ethif4**, and **ethif5**. Each port uses a RJ-45 jack that supports two Ethernet standards: *10Base-T Ethernet* i.e. Ethernet with bandwidth up to 10 Mbps and *100Base-T Ethernet* i.e. Fast Ethernet with bandwidth up to 100 Mbps. For both *10Base-T* and *100Base-T Ethernet* a pair of wires are used where one is for transmitting and the other for receiving the signal. Both standards supports half-duplex and full-duplex modes for transmission. Half-duplex means that only one can send while the other receives and full-duplex means that both can send and receive at the same time. The transmission speed and which duplex mode to use can be selected using auto-negotiation between connected devices. **efmif1** is the DSL port, which uses a RJ-11 Jack, and connects the MAG with the subscriber DSL access link to the ISP.

The Physical Wireless Interface

The physical wireless interface has the default port: **wlif1**. The physical wireless interface belongs to the internal access point component, which connect it to the bridge. The access point gathers the wireless signal by an antenna, transforms the signal into an Ethernet frame and forwards it to the bridge port through the appropriate physical wireless interface. The access point is capable of supporting up to four WLANs that results in four wireless ports with each port having a unique connection to a bridge port. The access point contains only one radio and that radio supports only one channel at a time, which means that all WLANs are represented on that single channel.

The state of the physical Ethernet and physical wireless interfaces can be determined with the CLI command => *eth device iflist*. The command displays the name, type and state of each port and is seen in Listing 3.1.

```
=> eth device iflist
Name         Type   MTU   MaxMTU   State
ethif1       FE     1526  1526     connected   100BaseTFD
ethif2       FE     1526  1526     enabled     Not connected
ethif3       FE     1526  1526     enabled     Not connected
ethif4       FE     1526  1526     enabled     Not connected
efmif1       --     1530  1984     connected   100M
wlif1        --     1526  1526     connected   130M
ethif5       --     1526  1526     enabled     100M
ndisveth0    --     1526  1526     connected   100M
```

Listing 3.1: Status of Physical Ethernet and physical wireless interfaces. *The list does not show autonegotiation settings.*

3.2.2 Ethernet Switch and Ethernet Bridge

The Ethernet switch and the Ethernet bridge are two layer 2 switches with functional differences: The Ethernet switch supports only basic forwarding with high speed, whereas the Ethernet bridge will forward frames with a lower speed, but support more advanced functions like Virtual Local Area Network (VLAN)s, priority tagging and so on. Figure 3.4 shows the placement and basic function of the two switches, where the red arrows signify a traffic flow.

The Ethernet bridge in the MAG is implemented based on the IEEE 802.1D standard, which defines the Media Access Control (MAC) bridge concept and its processes. The bridge is transparent so that a host connected to the network does not notice the presence of the bridge in the communication. The bridge is also self-learning, which means it contains and maintains a filtering database where all information needed for MAC forwarding is stored. The entries in the filtering database are dynamically learned by the bridge during a learning process and are based on the layer 2 frames received by the bridge. The forwarding process uses the database in order to forward frames through particular bridge ports or to drop received frames that do not belong.

VLANs and Multiple Bridge Instances

VLANs are based on the 802.1Q standard, which is a layer 2 technology used for network segmentation, enabling the possibility to create different layer 3 subnetworks over a single physical infrastructure. Devices belonging to one VLAN within a Local

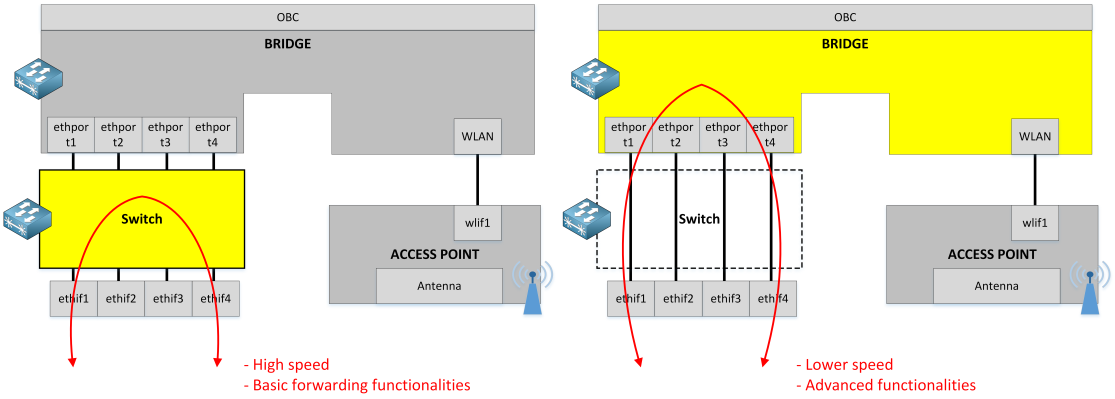


Figure 3.4: The Ethernet bridge and Ethernet switch forwaring processes. The ports seen in the figure will be explained in Section 3.2.3.

Area Network (LAN) can communicate with each other directly through the bridge, as if they were placed in a single and isolated LAN. In other words:

- Broadcast packets sent from a device in a VLAN, will reach all the devices within that VLAN, but will not reach devices belonging to the other VLANs
- Devices within one VLAN can communicate with each other using layer 2 principles. Meaning that they can send packets directly to other devices using MAC addresses and Address Resolution Protocol (ARP) messages, without the need to have the packet sent through the router instance.
- Communication between different VLANs is possible, but needs to be routed through the router. The router will then forward packets between different networks. This process is known as *InterVLAN* routing.
- Different VLANs differ using a VLAN Identifier (VID). The VID tag is unique for each VLAN and is used to identify VLANs throughout the MAG.

An alternative for doing network separation in the MAG is the creation of multiple bridge instances. Each bridge works as an independent instance and is connected to the router over different links. Thereby there is no direct connection between bridge instances and no Ethernet traffic can exist between them. Figure 3.5 presents the concept of multiple bridge instances. A newly created bridge supports VLANs and QoS just as in the ordinary bridge instance.

3.2.3 Bridge Ports

The (Ethernet) bridge ports are the ports belonging to the bridge that enables the bridge to link with the physical ports at layer 1, but also enables a link to higher layer interfaces. It is possible to create new bridge ports or modify already existing ports. The

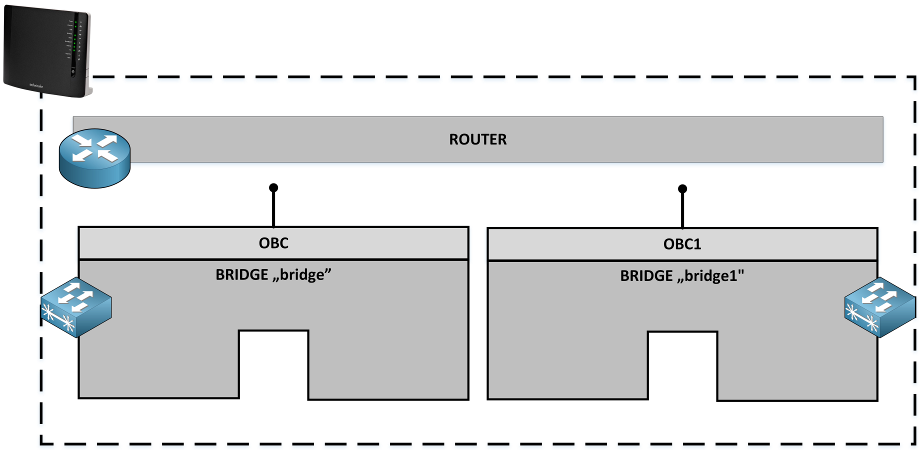


Figure 3.5: The concept of multiple bridge instances.

default bridge ports are **ethport1**, **ethport2**, **ethport3**, **ethport4** and **Wireless Local Area Network (WLAN)** as seen in Figure 3.2 on page 30. Bridge ports can connect to the following lower layer interfaces:

- **Physical Ethernet interfaces**: By default the bridge ports, listed above, are connected respectively to the Physical Ethernet interfaces: **ethif1**, **ethif2**, **ethif3** and **ethif4** as seen in Figure 3.2.
- **Physical wireless interfaces**: By default the **WLAN** bridge port is connected to the wireless physical interface **wlif1** as seen in Figure 3.2.
- **ATM interfaces**: The bridge ports can connect to ATM interfaces, but are not of interest in the network integrity part.

A special bridge port is the Onboard Controller (OBC) (see Figure 3.2). It is an internal port that manages and monitors the buffer between a bridge instance and upper layer interfaces. The OBC might be considered as a connection from the layer 2 bridge to layer 3. The bridge can be connected to particular upper layer interfaces (through the OBC) as follows:

- **Logical Ethernet interfaces**: A Logical Ethernet interface has to be configured with the bridge as a destination to a specific VLAN. A VLAN can posses only one Logical Ethernet interface.
- **LocalNetwork IP interface**: The **LocalNetwork** IP interface is directly connected to the bridge (more on this in Section 3.2.4).
- **PPP relay interfaces**: The bridge ports can connect to PPP relay interfaces, but are not of interest in the network integrity part.

In the case of multiple bridge instances, each bridge possess its own independent OBC port. Newly created bridges act like the default bridge. It means that new bridge ports will have to be created for the new bridge. Physical Ethernet interfaces or wireless interfaces might be detached from the default bridge and attached to the new bridge. Just like the default bridge, the new bridge might be a destination for a Logical Ethernet interface and through that be connected to the IP interface in the router instance.

3.2.4 Logical Ethernet Interfaces

The logical Ethernet interface is a software entity used to forward data between the bridge and the router components. Each VLAN requires a dedicated logical Ethernet interface to allow traffic between the VLAN in the bridge and an appropriate IP interface in the router. From the perspective of the bridge to the router, the logical Ethernet interface will only forward packets with a correct VLAN tag. From the perspective of the router to the bridge the logical Ethernet interface will insert an appropriate VLAN tag in the packet of the VLAN that it is assigned to and then forward the packet to the bridge.

The default logical Ethernet interfaces are **eth_wan** and **eth_wan_data**. As mentioned in Section 3.2.3, the **LocalNetwork** IP interface is connected directly to the bridge. This is possible because of an *invisible* Logical Ethernet interface between the bridge and the IP interface. Moreover, it can be applied to only one IP interface in the MAG such that this IP interface can omit the VLAN tag (receives and forwards untagged packets).

Logical Ethernet interfaces can be connected to the following lower layer interfaces:

- Lower layer:
    - **The bridge**: A logical Ethernet interface can be connected to the bridge as mentioned in Section 3.2.3.
    - **Physical Ethernet interface**: If a link between a bridge port and physical port is deleted, the Logical Ethernet interface can be directly connected to that physical port.
    - **Physical Wireless interface**: As with the physical Ethernet interface, a physical wireless port might be directly connected to a logical Ethernet interface.
    - **Logical Ethernet interface**: Logical Ethernet interfaces can be stacked on each other.

Logical Ethernet interfaces can be connected to the following upper layer interfaces:

- Upper Layer:
    - **IP interface**: A logical Ethernet interface can be connected to the Layer 3 IP interface.

- ◦ **Logical Ethernet interface**: Logical Ethernet interfaces can be stacked on each other.
- ◦ **PPP interface**: PPP interfaces are not of interest in this project.
- ◦ **Relay interface**: Relay interfaces are not of interest in this project.

3.2.5 IP Interfaces

An IP interface is a layer 3 interface[1] belonging to the router that is used to do layer 3 IP routing as well as *interVLAN* routing. In order for the router to route packets for multiple VLANs, it needs an IP interface for each VLAN. Each IP interface has an assigned an IP address and acts as a default gateway for devices connected to that specific VLAN. The default IP interfaces are **LocalNetwork**, **ip_wan_voip**, **ip_voip**, **internet_pppoa**, and **internet_pppoe** as seen in Figure 3.2 on page 30.

3.2.6 The Bridge and The Router Forwarding Process

For a better understanding of the forwarding process with the use of VLANs, logical Ethernet interfaces and IP interfaces, Figure 3.6 is presented. For the purpose of this example three computers are connected to the bridge ports **ethp1**, **ethp2**, **ethp3** through the respective physical interfaces **ethif1**, **ethif2** and **ethif3**. Individual VLANs have been created for each bridge port and associated with an IP interface through a logical Ethernet interface. VLAN:1 is the default VLAN and is associated with the "invisible" logical Ethernet interface. When a host sends a packet, an appropriate VLAN tag is inserted by the bridge according to the port the host belongs to. All packets are then sent to the OBC port, which distributes them to their appropriate logical Ethernet interfaces. The logical Ethernet interface sees if the VLAN tag is the same as it is assigned to and forwards it to its IP interface. The router strips off the layer 2 frame header and looks for an destination IP address that is compared with the routing table. It is done in order to make a decision of where to route the packet.

In the situation when packets are forwarded from the router, the router forwards packets to an appropriate IP interface. The IP interface will then forward the packets to its logical Ethernet interface, which will assign the VLAN tag and forward packets to the bridge. The bridge will forward the packets to the appropriate port and host.

3.3 Networking Services

Router component in the MAG includes functions like NAT, DNS, DHCP, and Firewall. These elements are also part of the MAG and are no less important than the relationship between interfaces. An understanding of each of them and their purpose is necessary:

[1] Services like firewall, NAT, DHCP, and DNS belongs to the router component and not the IP interface.

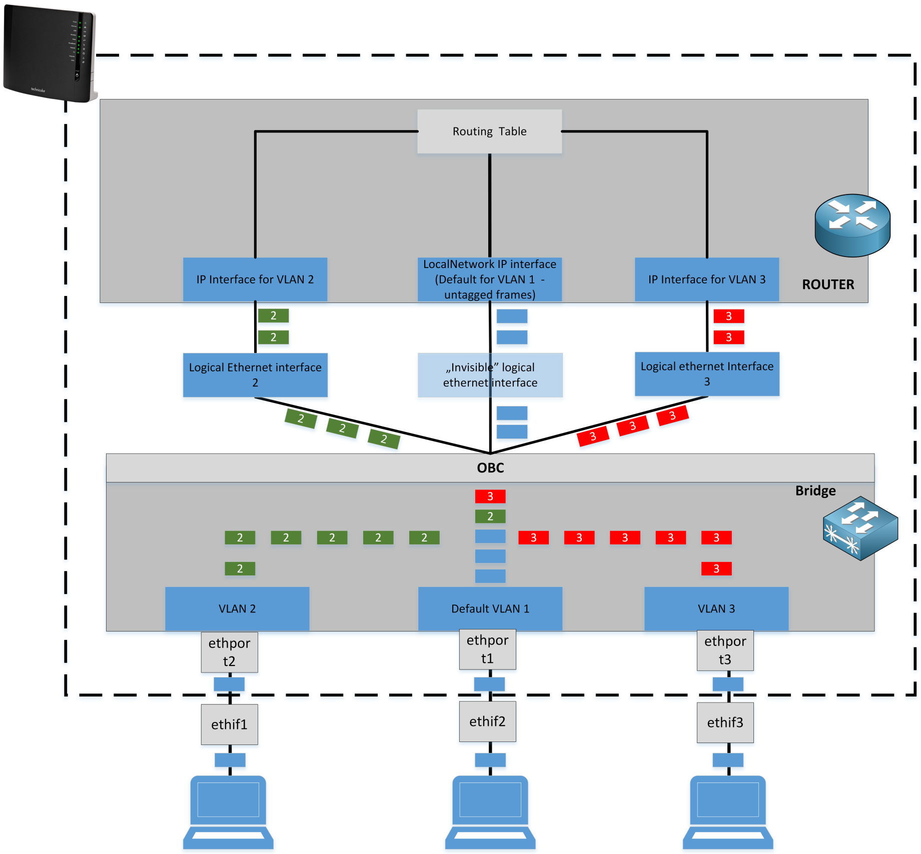

Routing Table
IP Interface for VLAN 2
LocalNetwork IP interface (Default for VLAN 1 - untagged frames)
IP Interface for VLAN 3
ROUTER
Logical Ethernet interface 2
„Invisible" logical ethernet interface
Logical ethernet Interface 3
OBC
Bridge
VLAN 2
Default VLAN 1
VLAN 3
ethport2
ethport1
ethport3
ethif1
ethif2
ethif3

Figure 3.6: The forwarding process from the bridge to the router.

3.3.1 Network Address Translation (NAT)

By the end of 2016 it is expected that 6.4 billion devices will be connected to the Internet. When using IPv4 it is possible to assign (at most) $2^{32} = 4,294,967,296$ (4.3 billion) unique IP addresses (not considering invalid and reserved addresses). It should be obvious to see the problem[2]. The solution to the problem is NAT, which translates several private IP addresses into a single public IP address. The principle is shown in Figure 3.7.

The purpose of this translation is to let several clients, belonging to one network e.g. in a modem, use a single public IP address. This is useful when modems can reuse the same private IP addresses for clients, but each modem has a unique public IP address that can be recognised by Internet routers. To be able to do this the NAT (in each modem) has to keep track of connections created by each client in the subnet as well as the translations. This translation process is shown in Figure 3.8 on page 39. It

[2] It is the one (if not the most) obvious reason to why the IPv6 header was introduced ($2^{128} = 3.4 \cdot 10^{38}$ unique addresses). This project only considers IPv4 traffic as Telenor has not implemented IPv6 traffic yet (18th February 2016).

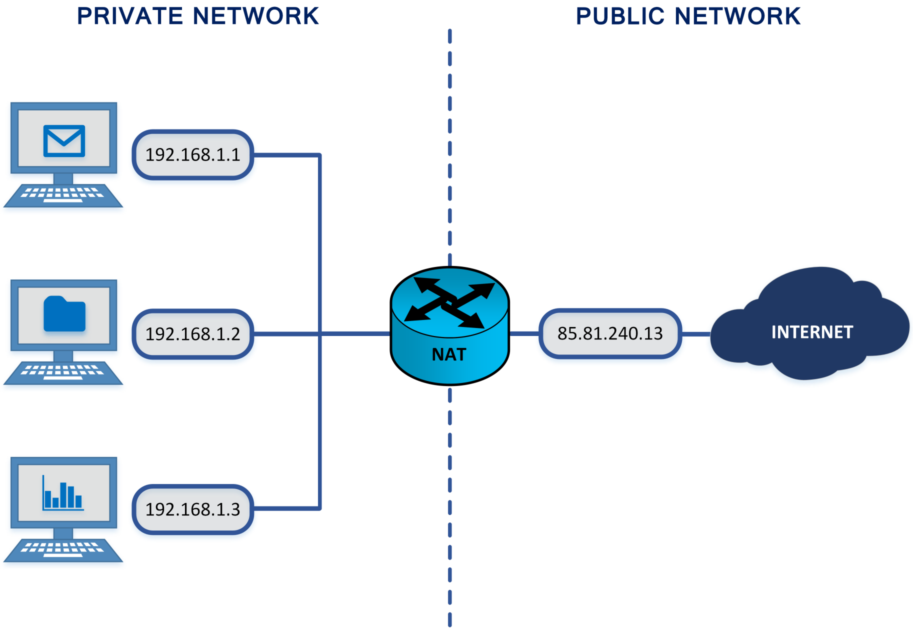


Figure 3.7: How three IP addresses in a subnet are translated into a single public IP.

shows how a client (`192.168.1.1`) sends some data to another public IP address (`188.172.21.13`). The IP address of the client is translated by the NAT (`85.81.240.13`), such that it is possible to receive a reply from `188.172.21.13`. If `188.172.21.13` were to reply to `192.168.1.1` the message would be discarded or lost. The NAT does not only translate IP addresses, but also the port used (Network Address and Port Translation (NAPT)). If several clients in a subnet were to use the same source port it would cause conflicts when the NAT receives a reply, which is why the port is also translated.

NAT in the MAG

The NAT in the MAG has three modes:

Enabled: Translates the IP address (and port) of a client into the public IP address and another port just as described.

Disabled: Address translation is disabled.

Transparent: When NAT is in *transparent* mode it is somewhere between the *enabled* and *disabled* modes. NAT is not performed for internal devices that do not allow any address translation due to strict protocol rules e.g. IPSec. When using *transparent* NAT, it is needed to add extra NAT configuration.

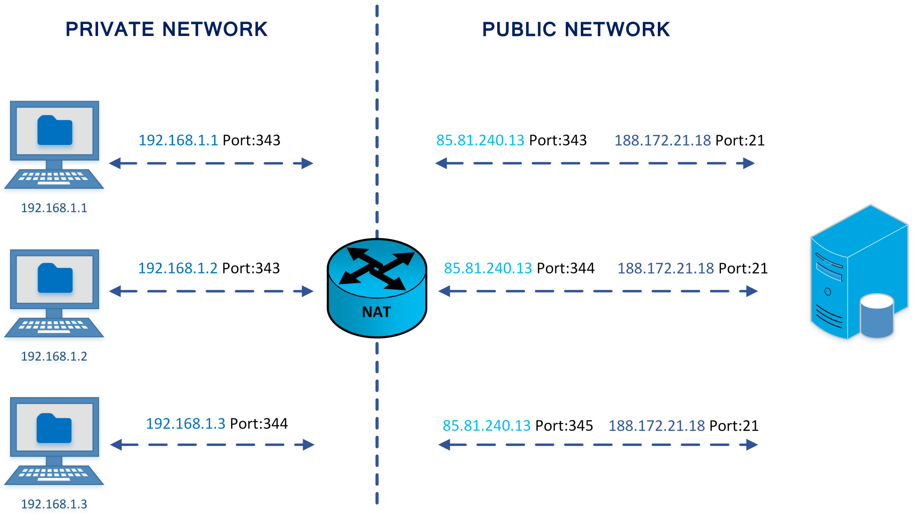


Figure 3.8: How a single IP address in a public is translated into a public IP address, while the NAT remembers the translation.

3.3.2 Domain Name System (DNS)

The function of DNS is to associate domain names (e.g. web servers) with IP addresses [Notenboom [2013]][3]. The process is very fast, but includes communication with four different servers to go from a domain name to an IP address (See Figure 3.9). The function of each server included in the process is [Decker [2014]]:

1. **Resolving Name Server**: Is a local DNS server typically located at the ISP or in the modem. This server can cache previous DNS lookups. If the DNS is not cached the Resolving Name Server will refer to a Root Name Server.
2. **Root Name Server**: Knows which servera are responsible for `.net`, `.com`, `.org` domains etc.
3. **TLD Name Servers**: Each server is responsible for either `.net` or `.com` or `.org` domains etc. This server does not know the the correct IP but can refer to a server that does know the IP of the requested domain.
4. **Authoritative Name Servers**: This server knows the IP of the domain (and subdomains) requested.

The process can be either iterate, recursive or a mix of both. In an iterate process the response from each server will go back to the client, who will then ask the next server. In a recursive process each server will forward its answer to the next server in the list. The last server (Authoritative Name Server) will then send an answer back, which will recursively go all the way back to the client with an IP address.

[3] The process can also be reversed (reverse DNS) to gather a domain name from an IP address. This can e.g. be useful to check the identity of a domain name.

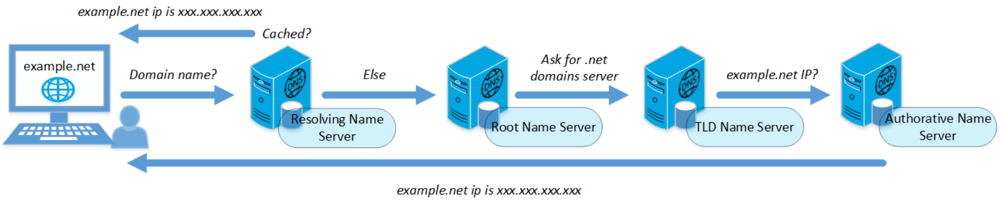


Figure 3.9: The DNS process that involves the four DNS servers.

The local DNS Server of the MAG

The MAG itself has its own local DNS server (*DNS-S*) which can cache DNS lookups. If a DNS lookup is not cached by the local DNS server it will forward the request to the Resolving Name Server located at Telenor. From this point the local DNS server will get the answer iteratively or recursively, while the client is waiting for an answer (recursive between client and local DNS server).

3.3.3 Dynamic Host Configuration Protocol (DHCP)

If IP addresses are statically assigned, each client will have to manually fill in a unique IP address. If two clients are using the same IP address they will have conflicting IP addresses, where at least one would have to disconnect or fill in another IP address. The problem might not be troublesome for very small size networks, but as soon as the network grows it will be impractical to keep track of assigned IP addresses manually. The problem can easily be solved by introducing dynamic assigned IP addresses.

The function of a DHCP server is to maintain and dynamically lease out IP addresses for connected/connecting hosts. The DHCP server has an IP pool from which is can assign IP addresses. To do this it stores information about the gateway address, DNS server address (to offer clients), a list of connected clients, and a list of leased clients. When a client requests a lease it uses a process known as DORA (Discover, Offer, Request, Acknowledge), which consists of four steps[Encyclopedia [2016]][Microsoft [2016a]]:

1. **DHCPDISCOVER**: The clients broadcasts (`255.255.255.255`) a DHCP discover request for a DHCP server from source `0.0.0.0` to port `67`.
2. **DHCPOFFER**: All DHCP servers, that hear the *DHCPDISCOVER* request, broadcast a *DHCPOFFER* consisting of network parameters offered to the client. The parameters offered are e.g. gateway address, subnet mask, lease time, new client IP address as well as a DHCP server IP address.
3. **DHCPREQUEST**: The client will respond by broadcasting a *DHCPREQUEST* (request a lease) with the server information of the server it wishes to belong to from source IP `0.0.0.0`.

4. **DHCPACK**: Finally the DHCP server can acknowledge the client's request by broadcasting an acknowledgement from the DHCP server's IP address. The DHCP will update its server information to include the newly assigned client.

DHCP Lease

When a client is connected it has only leased its IP address for a specified amount of time. When the lease is about to expire it should be renewed if the client wishes to keep the IP address, which is in most cases highly desired. If a client does not renew the lease it *will* lose the connection to services it might be using. The NAT will e.g. not be able to map a packet to the client if its IP address has suddenly changed. When the DHCP lease time reaches less than 50 percent of its total assigned lease time the client will have to renew the lease (if possible). It takes only two steps to renew the lease and it is done by sending a *DHCPREQUEST* and then expect a *DHCPACK* from the DHCP server [Microsoft [2016b]]. It is also a possibility that the DHCP server will decline the request or offer a new assignment, which the client will have to take care of. The DHCP server does not care about the client and its lease time even though it keeps track of how much time is left for each leased client. The DHCP server keeps track of lease times such that it can pass on an IP address to another client, which is useful in crowded networks. The total lease time highly depends on the network e.g. how big is the network (subnet mask), how many clients use the network (during a lease time), will clients reconnect later etc. Imagine how to define lease times for a hotel, a restaurant, or a work station.

Relaying Agent

A relaying agent is needed in networks with multiple subnets. Its function is to relay DHCP messages between DHCP clients and DHCP servers to let a DHCP client connect to a specific IP subnet [Wikipedia [2015]] [itgeared [2012]]. When a client broadcasts a DHCP request, the relaying agent will receive that broadcast, transform it into a unicast message (the router do not forward broadcast messages) and then forward it to the intended DHCP server. Explained in another way: When a client broadcasts a *DHCPDISCOVER* it is meant for one subnet, but it does not have a specific destination yet. The message is collected by the relaying agent at `127.0.0.1` and if a router has more than one subnet it should notify the DHCP server for the intended subnet such that it can broadcast a *DHCPOFFER* for the client. The task for the relaying agent is to know which subnets are available and notify the subnets when *DHCPDISCOVER*'s or *DHCPREQUEST*'s are received.

3.3.4 The Firewall within the MAG

The firewall in the MAG classifies packet flows to chains in the following way:

forward: Traffic from an interface to another interface.
source: Traffic from the MAG
sink: Traffic to the MAG

If there is a wish to define rules between interfaces the *forward* packet flow is used. An interface is for example a device connected to the MAG or the **WAN** group i.e. the group defined for outgoing traffic. In the *forward* packet flow, it is possible to add firewall rules in the following chains:

forward_level_BlockAll: Block all traffic to/from the Internet.
forward_level_Standard: Allow outgoing connections and block incoming traffic.
forward_level_Disable: Disable the firewall. Allow all traffic.

Each rule is associated with a chain, an index, a name, an action and a condition each having the following properties:

- **Chain**: When a packet is going through the firewall it is first classified into its appropriate chain.
- **Index**: The firewall goes through the rules in sequential order from index 1 and finishes at first match.
- **Name**: The name does not have any function.
- **Action**: An action could be to *drop*, *deny* or *accept* a packet.
- **condition**: The condition, consists of **srcintf**, **srcip**, **dstintf**, **dstip**, and **serv**. It is the options to which a rule can comply, and to which the action applies. The condition entry has 4 entries and looks like: *serv 1.2 > 3.4* where:

  >: Can be read as *going to*.
  1: To which source interface the rule applies.
  2: To which source IP address(es) the rule applies.
  3: To which destination interface the rule applies.
  4: To which destination IP address(es) the rule applies.
  serv: To which services the rule applies to.

  Each of the entries 1, 2, 3 or 4 can use the wildcard *, which means '*for all*'. Conditions can also be made for different services e.g. TCP, ICMP, or a port. The rule *ftp lan.* > *.192.168.1.1* will e.g. apply for FTP traffic (port: `21`) from *lan* going to IP `192.168.1.1`. The default services in the MAG can be found in Listing D.3 on page 215 in Appendix D.

The default firewall rules cannot be shown directly in the report, but can be found in Appendix D.1 on page 210.

3.4 Wireless Security Methods

So far, the sections in this Chapter has described specific internal components, architecture and services of the MAG. However, while configuring the Service Set Identifier (SSID) it is required to define a security protocol that is intended to use. That is why the following section will focus on possible wireless security protocols supported by the MAG. Based on differences between the protocols the right one can be chosen for future solution design.

3.4.1 Security Protocols

Various protocols and technologies are used in wireless networks to create a mobile and stable connection with the wired infrastructure. That make wireless networks very complex. However, it is known that wireless networks are not as secure as wired networks where connection take place only between two end points connected by a cable. The wireless networks broadcast data to every nearby and listening device as well as in each direction within an specified range. In order to secure user data and prevent of unauthorized access different security protocols has been developed and are use in today's world. There are three major security standards for wireless networks:

- **Wired Equivalent Privacy (WEP)** -Encryption protocol created for wireless networks to ensure the same security as in wired networks. It is using RC4 encryption know also as stream cipher with 40-bit or 140-bit key. However, the key has to be manually typed into each device and access point without the possibility to change or rotate it. This well-known flaws makes it easily to brake. That is why in 2003 it has been replaced and not recommended to use anymore.
- **Wi-Fi Protected Access (WPA)** - security protocol and certification developed by the Wi-Fi Alliance in order to create more secure wireless networks compared to the weak WEP protocol. It uses Temporal Key Integrity Protocol (TKIP) which is also a RC4 encryption, however it dynamically generates a new 128 bit key for every new packet which make it secure for the types of attack used in WEP.
- **Wi-Fi Protected Access version 2 (WPA2)** - IEEE 802.11i replaced WPA in 2004. Main improvement over WPA protocol is that WPA2 abandoned TKIP algorithm that because of known vulnerabilities. WPA2 is using Advanced Encryption Standard (AES) a 128, 192 or 256 bits symmetric-key length (same key is used for encrypting and decrypting data) that is used to encrypt classified and sensitive data. Each new device requires a mandatory WPA2 certification.

Both WPA and WPA2 use either WPA-PSK (Pre-shared key) or WPA-Enterprise as authentication methods. The WPA-Enterprise method requires a RADIUS server. The function of the RADIUS server will be introduced in Part III.

The next chapter will go through the requirements specified for this Part of the report. After the requirements a proposed solution design will be presented, which uses the knowledge gathered from this chapter to draw design conclusions.

CHAPTER FOUR

SOLUTION DESIGN

This chapter will provide a solution design to network integrity, based on the requirements and the analysis part of the MAG described in previous chapter.

The requirements for network integrity are constructed so that, after fulfilling them, a right level of privacy should be ensured, as mentioned in Chapter 3. The requirements are as follows:

Network Integrity

Must have

1. Hotspot users will not have access to any devices in the private network.
2. Hotspot users will not have access to each other.
3. Hotspot users will not have access to services in MAG.
4. Private users will have access to each other.
5. Private users will not have access to the anything in the hotspot network.
6. Private users will be able to access services in MAG.
7. Hotspot users will have access to the Internet.
8. It will not be possible to access hotspot devices from the public Internet.
9. Access to the Internet for private users will not be changed.
10. Hotspot users will have the same accessibility to the public FTP server as anyone else.

Should have

11. Users in the hotspot network will not be able to see if a devices in the hotspot network are connected or not.
12. Users in the hotspot network will not be able to see if a devices in the private network are connected or not.
13. Users in the private network will be able to see if a devices in the private network are connected or not.
14. Users in the private network will not be able to see if devices in the hotspot network are connected or not.
15. Hotspot users should not be able to get information of devices that have been or are connected.
16. Private users should not be able to get information of devices that have been or are connected to the hotpsot network.
17. Hotspot users will use the best wireless security available in the MAG.
18. The DHCP should only distribute IPv4 addresses to hotspot users.

4.1 System Design

There are several ways to meet the requirements for network integrity. This section will describe the solution design and choices made. The requirements state that there needs to be a clear separation between users connected as private users or hotspot users. An intuitive way to do this is to separate users into two independent networks i.e. a network for private users and a network for hotspot users. The MAG supports multiple VLANs and multiple bridge instances, both of which can be a solution. Section 3.2.2 describes both and a decision has been made to create another VLAN and a second wireless access point for the hotspot network. The reason to this is that, even if another bridge instance is created, it will still require the creation of a VLAN for that bridge instance. To our knowledge, this design choice does not introduce any vulnerabilities, because traffic from one VLAN to another will always go through the router instance on equal terms.

The wireless security protocol used for the hotspot network will be the WPA2 protocol, because it is the most secure protocol at the moment and is also the one used for the private network.

The previous chapter introduced an extensive description of the interface architecture of the MAG and gave some insight into how to implement a hotspot network. It requires a new physical wireless port, a bridge port, a Logical Ethernet interface, and an IP interface. Figure 4.1 illustrates the location and connections of the new interfaces in the MAG (marked in a light blue color) and can be compared to the default interfaces.

The next chapter will describe the implementation in order to meet the requirements, and to create a solution that maintains the Network Integrity of all users.

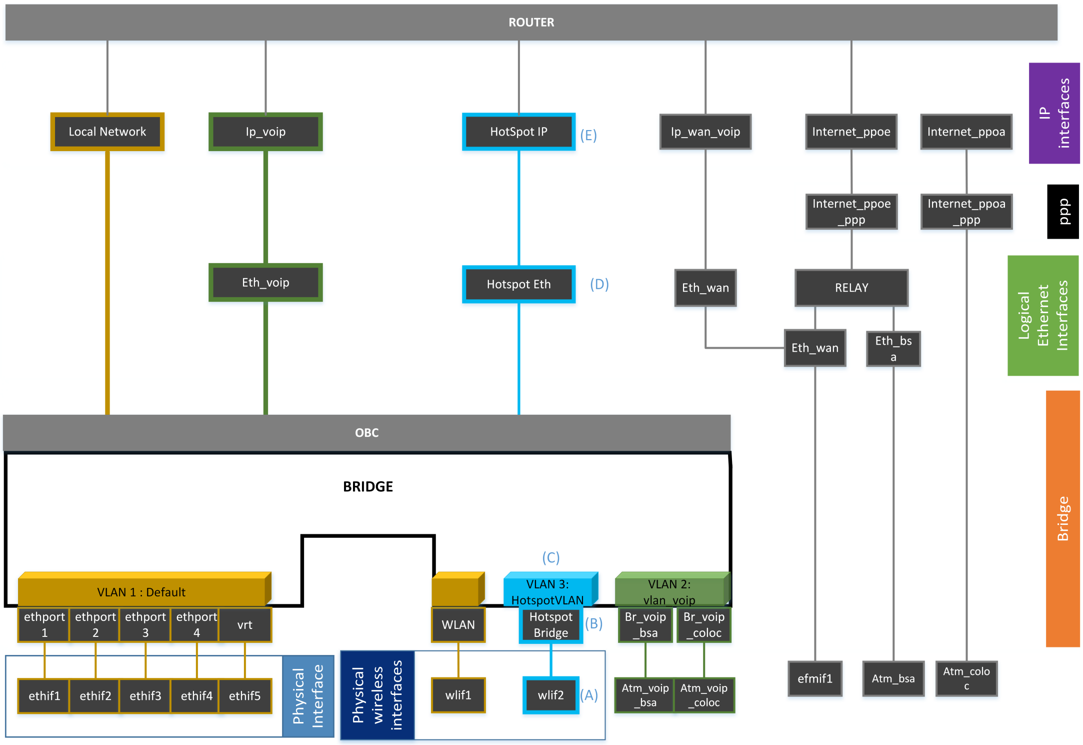


Figure 4.1: Proposed solution design

CHAPTER

FIVE

IMPLEMENTATION

Chapter 3 went through an analysis of the internal structure of the MAG and available security methods. To ensure a high level of security it is decided that the wireless hotspot network will use WPA2, which uses AES, as the encryption protocol. WPA-Enterprice will authenticate each user using the EAP-SIM protocol, as each user is a mobile subscriber of Telenor. Hotspot users should not, in any way, have access to the private (local) network of the paying xDSL subscriber of the MAG and it is therefor decided to make an isolated VLAN only for the hotspot network. It is of high importance that hotspot users do not get unauthorised access to services as for example the *MediaAccess Gateway GUI* or any *Telnet services*. Access to these should only be available to the private user who is the paying subscriber.

As far as it is known the MAG can be accessed and configured in four ways:

- Using the *MediaAccess Gateway GUI*, that can be accessed with the *HTTP protocol* on port `2033` from the LAN by typing `http://10.0.0.1:2033/` in the browser.
- Through a *Telnet session* at port `23023` that can be accessed from anywhere.
- Telenor can modify the MAG through a *HTTPS session* on the public IP of the modem at port `2034`.
- *TR-069* is an application layer protocol that uses the *HTTP protocol*. It enables bidirectional communication between CPEs and Auto Configuration Servers (ACSs). It enables automatic and secure configuration as well as control of CPEs [John Blackford [2013]].

All modifications requires a username with matching password. The documentation and the opportunity to make configurations through the *MediaAccess Gateway GUI* is somewhat limited, which is why the setup will be done through the Telnet service at port 23023. This Chapter will document each step in the configuration of the MAG to make a secure hotspot network for Telenor mobile subscribers.

Each step of the configuration will be documented with the configuration commands and a list of the settings will usually be shown before a configuration is made and again when a configuration is complete. The reference [NPR [2015]] is used as a source of inspiration for this Chapter.

5.1 Configuration of the Wireless Hotspot Network

This Section is based on the solution proposal given in Section 4.1. The purpose is to create a wireless SSID for hotspot users and move this SSID to a new *VLAN* in a different subnet than the IP interface of the private network.

The configuration will consist of the following steps and is based on the setup shown in Figure 4.1 on page 47:

Layer 1 **(A)** Create a physical layer 1 port (WLAN) with a SSID for the hotspot network.

Layer 1→2 **(A+B+C)** The newly created WLAN will be assosiated with a layer 2 bridge port which is a member of a specific VLAN. The bridge port should not belong to the VLAN used by the private network for security reasons, so that a new VLAN must be created. When the VLAN is created it belongs to the layer 2 *bridge* and has the bridge port as a member.

Layer 2→3 **(C+D+E)** The VLAN now belongs to the *bridge* and will need a layer 3 IP interface. The IP interface connects to a layer 2 logical Ethernet interface, with tagged frames going from the IP interface to the bridge and from the VLAN to the IP interface. Using that tag, the bridge can forward frames from the IP interface to the VLAN and vice versa.

Layer 3 **(E)** A subnet and a gateway address should be defined for the IP interface. NAT should be disabled for the IP interface as no address translation is needed within the subnet.

Layer 3 **(E)** The subnet for the IP interface should be known by the DHCP server. This is done by adding it to the DHCP server pool of the relaying gateway.

Each step in the configuration has its own section, explaining the steps taken to configure the wireless hotspot network.

5.1.1 Create a WLAN SSID for Hotspot Users

The first step in the configuration is to create a wireless hotspot SSID which is seen as letter **A** in Figure 4.1. The new SSID has to be created to ensure that hotspot users will access another wireless network other than the one private users will access. A *public* wireless SSID will be created and it is decided to give it the following setup as of Chapter 3:

SSID: *TelenorHotspot*

Client isolation: *Enabled*[1]

Public network: *Enabled. any* can be used to show/hide the SSID.

Security mode: *WPA-Enterprise*

WPA version: *WPA2*

RADIUS IP: `192.38.55.78`[2,3]

RADIUS port: `20012`

RADIUS key: "*radiusSecret*" *(Should be changed and only known to Telenor)*

Radio ID: *0*[4]

Max associations: *8*

With the protocols available for WiFi in the MAG, these settings will ensure a high degree of security: Clients cannot *see* each other, the network is listed as a public network and it uses WPA2[5]. Listing 5.1 shows a list of the default SSIDs on the router. It can be seen that only the private SSID for the xDSL subscriber is configured at this point.

```
=> wireless mssid iflist
ssid 0/0 : Telenor94467B    [up]    Security: WPA-PSK
```

Listing 5.1: A list of the default SSID(s) of the router.

Listing 5.2 on the next page is the configuration that set up the hotspot SSID: **TelenorHotspot** and Listing 5.3 on the following page is a list of WLANs when the configuration has been made. During the configuration the router gives the following: *Allocated ssid id[0/1] ethernet port[wl_ssid1_local0]* which is the physical port identification: **wl_ssid1_local0** i.e. *SSID: 1* & *radio ID: 0*. This physical port identification will later be used to associate the WLAN with a bridge port.

[1] APisolation or client isolation is mostly used for public networks to make sure that clients connected to a network can not *see* each other. It adds extra security as the link between clients is deleted as well.

[2] In order to use WPA-Enterprise authentication, a RADIUS server is needed as will be explained in Section 14.1.2. The RADIUS server has an IP address, a port and a secret key.

[3] The IP address, port, and secret key for the RADIUS server will be defined in Section 16.1.

[4] Radio ID can also be expressed as antenna identification and can be used if multiple antennas are available. The MAG has only one radio (with two antennas) available with index Radio ID: *0*.

[5] WPA2 uses AES to encrypt the wireless transmission of data.

```
=> wireless mssid ifadd ssid=TelenorHotspot
    Allocated ssid id[0/1] ethernet port[wl_ssid1_local0]
=> wireless mssid ifconfig ssid_id=1 ssid=TelenorHotspot
    apisolation=enabled any=enabled secmode=wpa WPAradiuskey="
    radiusSecret" WPAradiusip=192.38.55.78 WPAradiusport=20012
    WPAPSKversion=WPA2 WPAversion=WPA2 radio_id=0 maxassociations=8
=> wireless mssid ifattach ssid_id=1
```

Listing 5.2: Set up **TelenorHotspot** SSID with WPA2-Enterprise authentication.

```
=> wireless mssid iflist
ssid 0/0 : Telenor94467B    [up]    Security: WPA-PSK
ssid 0/1 : TelenorHotspot   [up]    Security: WPA
```

Listing 5.3: The list of SSIDs when the configuration has been done. The private SSID is seen along with the newly created **TelenorHotspot** with WPA(2)-Enterprise security.

As the MAG only has one radio it is not possible to give **Telenor94467B** and **TelenorHotspot** different wireless channels. The MAG periodically scans the wireless medium to select the best channel available i.e. the channel with the least amount of interference.

5.1.2 Implementation of a Virtual LAN

To improve the security between the private network and the hotspot network, the hotspot network should have a seperate VLAN. The VLAN is a layer 2 protocol that divides clients into different logical subnetworks. When hotspot users are members of another VLAN they cannot see traffic from the private network and vice versa.

In this step of the configuration a new VLAN will be created. This VLAN will belong to the layer 2 **bridge** seen in Figure 4.1. The VLAN will have a port member called **HotspotBridge** (bridge port), which is associated to the WLAN created in the first part of the configuration. The **bridge** acts like a switch that supports VLANs. The **bridge** associates the layer 1 WLAN as a member of the layer 2 VLAN using the bridge port obtained from Listing 5.2. Listing 5.4 shows the four VLANs of the **bridge** each having a unique VLAN Identifier (VID).

```
=> eth bridge vlan iflist
Vid   Name         Bridge interfaces (* = untagged)
---   ----         -------------------------------
1     default      OBC*, ethport1*, ethport2*, ethport3*, ethport4*,
                    WLAN*, virt*
2     vlan_voip    OBC, br_voip_bsa*, br_voip_coloc*
34    vlan_eth_voip
101   vlan_eth_data
```

Listing 5.4: The default VLANs of the **bridge**.

The configuration of the hotspot VLAN can be seen in Listing 5.5. It shows that it connects the physical bridge identifier i.e. **wl_ssid1_local0** to the bridge port **HotspotBridge**. It attaches **HotspotBridge**, which becomes part of the **default** VLAN. **HotspotBridge** is then connected to **HotspotVLAN**, while the association to **default** is deleted, which finishes the implementation of layer 1 to layer 2.

```
1 => eth bridge ifadd intf=HotspotBridge dest=wl_ssid1_local0
2 => eth bridge ifattach intf=HotspotBridge
3 => eth bridge config vlan=enabled
4 => eth vlan add name=HotspotVLAN vid=3 addrule=disabled
5 => eth bridge vlan ifadd name=HotspotVLAN intf=OBC untagged=
     disabled
6 => eth bridge vlan ifadd name=HotspotVLAN intf=HotspotBridge
     untagged=enabled
7 => eth bridge vlan ifdelete name=default intf=HotspotBridge
     Warning: default VLAN changed for bridge interface to VLAN
     HotspotVLAN.
```

Listing 5.5: Associate the newly created **TelenorHotspot** SSID to a new VLAN **HotspotVLAN**.

Line 1: Create a bridge port interface named **HotspotBridge** and associate it with the port identifier for the WLAN.

Line 2: Attach **HotspotBridge**. It will be attached to the **default** VLAN.

Line 3: Let the **bridge** interpret VLAN tags by making it VLAN aware.

vlan: If this option is *enabled* it makes the **bridge** VLAN aware. It indicates if the **bridge** interprets the tag of VLAN frames. If the **bridge** is VLAN aware it will forward the frames to the correct bridge port.

Line 4: Add a new VLAN **HotspotVLAN** with VID=*3*. The VLAN will not have a shared filtering database with the **default** VLAN.

VID: *VID* is the unique VLAN identifier that is used to tag frames.

addrule: *addrule* indicates whether the new VLAN will have a shared filtering database with the **default** VLAN or not. If *addrule* is *disabled* a new database will be created, which is recommended for new VLANs.

Line 5: All VLANs have to have the **OBC** (On-Board Controller) bridge port as a member it will be *untagged=disabled*, because it is already a tagged member by **default**.

untagged: Determines whether the tag of frames that come out of the bridge port will be removed.
If *untagged* is *enabled* the VLAN tag will be removed before the tag is sent by the bridge port. Each bridge port has to have an untagged member of at least one VLAN.
If *untagged* is *disabled* the tag will remain in the frame that was sent by the bridge port.

Line 6: Add **HotspotBridge** to the **HotspotVLAN** and let it be *untagged=enabled* because each bridge port has to have an untagged member of at least one VLAN.

Line 7: Finally delete **HotspotBridge** from the **default** VLAN as it now is a member of **HotspotVLAN**.

The result can be seen in Listings 5.6 & 5.7, whereas the latter further illustrates the connection between interfaces. All changes, from this point, will be made at the hotspot VLAN **HotspotVLAN** such that no changes should be made to the private network. This will provide user satisfaction, as private users can keep their custom MAG setup, without the new configuration compromising security or other settings on the private network.

```
=> eth bridge vlan iflist
Vid   Name        Bridge interfaces (* = untagged)
---   ----        --------------------------------
1     default     OBC*, ethport1*, ethport2*, ethport3*, ethport4*,
                   WLAN*, virt*
2     vlan_voip   OBC, br_voip_bsa*, br_voip_coloc*
3     HotspotVLAN OBC, HotspotBridge*
34    vlan_eth_voip
101   vlan_eth_data
```

Listing 5.6: **TelenorHotspot** SSID is now associated to **HotspotVLAN** VLAN using **HotspotBridge** bridge port. The default VLANs are not changed.

```
=> eth bridge iflist
[...]
HotspotBridge:  dest            : wl_ssid1_local0
                Connection State: connected     Retry: 10
                Priority Tagging: Disabled
                PortState       : forwarding    Interface: up
                PortNr          : 9
                multiWANuntagged: disabled
                Multicast filter: disabled
                Wan             : disabled
                IGMP snooping   : enabled
                MLD snooping    : enabled
                Transparent Prio: disabled
                BPDU Filtering  : disabled
                Extra Tagging   : none
                Dynamic VLAN    : disabled
                VLAN: Default VLAN: HotspotVLAN Ingressfiltering:
                                                  disabled
                    Acceptvlanonly: disabled
                    Priority Config: disabled   IP Prec: disabled
                          Priority: 0
                    Regeneration table  : 0 1 2 3 4 5 6 7
                RX bytes: 1266372       frames: 9404
                TX bytes: 179692        frames: 2252
```

Listing 5.7: Shows how **wl_ssid1_local0** (physical port identifier) is associated to **HotspotBridge** which is associated to **HotspotVLAN**.

5.1.3 Route VLAN to IP Interface

In Section 5.1.2, the bridge port was connected to the VLAN using the **bridge** at layer 2. This Section will connect the **HotspotVLAN** VLAN to a layer 3 IP interface **HotSpotIP**. To make this connection, a logical Ethernet interface is needed i.e. **HotspotEth**. The function of **HotspotEth** is to tag frames from the IP interface such that the **bridge** can see the intended VLAN i.e. **HotspotVLAN**. **HotspotEth** will not tag frames coming from **HotspotVLAN**, as they are already tagged in the **bridge**, and they will end up at **HotSpotIP** through **HotspotEth**.

Listing 5.8 shows the IP interfaces of the router before a new IP interface is allocated for the hotspot network.

```
=> ip iflist
Interface          Group   MTU     RX       TX          Admin    Oper
---------          -----   ---     --       --          -----    ----
1 loop.......... local   4096    12 MB    9398 KB     UP       [UP]
2 internet_pppoa wan     1500    0        0           DOWN     DOWN
3 internet_pppoe wan     1500    0        0           DOWN     DOWN
4 ip_voip....... wan     1500    0        0           DOWN     DOWN
5 LocalNetwork.. lan     1500    0        0           UP       [UP]
6 ip_wan_voip... wan     1500    0        0           DOWN     DOWN
```

Listing 5.8: The list of default IP interfaces.

Listing 5.9 is the configuration that creates **HotspotEth** and connects it to the **HotspotVLAN**. It then creates **HotSpotIP** and connects it to **HotspotEth** which finalises the connection. **HotSpotIP** it set to belong to the group: **dmz** such that it is possible to implement firewall rules for the whole group later. It is decided to disable *IPv6* for **HotSpotIP** because only *IPv4* will be used in this configuration . It will be disabled to overcome potential security issues[6].

```
1 => eth ifadd intf=HotspotEth
2 => eth ifconfig intf=HotspotEth dest=bridge vlan=HotspotVLAN
3 => eth ifattach intf=HotspotEth
4 => ip ifadd intf=HotSpotIP dest=HotspotEth
5 => ip ifconfig intf=HotSpotIP group=dmz ipv6=disabled
6 => ip ifattach intf=HotSpotIP
```

Listing 5.9: Configure the logical ethernet interface **HotspotEth** to connect **HotspotVLAN** to the **HotSpotIP** IP interface.

Line 1: Add a logical ethernet interface called **HotspotEth**

Line 2: Configure the logical ethernet interface to have a connection to the **bridge** and give frames the tag that routes to **HotspotVLAN**.

Line 3: Activate **HotspotEth**.

[6] From [Sandal [2014]] it can be read (in the comments) that YouSee might have had security problems regarding IPv6 traffic.

Line 4: Add a IP interface called **HotSpotIP** that connects to the logical ethernet interface **HotspotEth**.

Line 5: Configure the IP interface to be in the group *dmz* and disable IPv6.
The group is added such that it is easy to implement firewall rules and QoS later. IPv6 is disabled because it is not supported globally by Telenor[Stahl [2015]]. It is not wished to implement IPv6 as it does not add any functionality.

Line 6: Activate the IP interface **HotSpotIP**.

The result can be seen in Listing 5.10 which shows that a new IP interface named **HotSpotIP**. Listing 5.11 shows that the **HotspotEth** belongs to **HotspotVLAN** and its lower layer connection is **bridge**.

```
=> ip iflist
Interface         Group   MTU     RX      TX          Admin   Oper
---------         -----   ---     --      --          -----   ----
1 loop.........   local   4096    12 MB   9398 KB     UP      [UP]
2 internet_pppoa  wan     1500    0       0           DOWN    DOWN
3 internet_pppoe  wan     1500    0       0           DOWN    DOWN
4 ip_voip.......  wan     1500    0       0           DOWN    DOWN
5 LocalNetwork..  lan     1500    0       0           UP      [UP]
6 ip_wan_voip...  wan     1500    0       0           DOWN    DOWN
7 HotSpotIP.....  dmz     1500    377 KB  103 KB      UP      UP
```

Listing 5.10: The **HotSpotIP** IP interface is now allocated in the group: *dmz* and has state: *UP*.

```
=> eth iflist
[...]
HotspotEth   :    Dest: bridge
                  Connection State: Connected     Retry: 10
                  Wan: Disabled    Administrative MTU: 1500
                      Operational MTU: 1500
                  Priority Tagging: Disabled
                  PortNr: 6
                  VLAN: HotspotVLAN
```

Listing 5.11: Connection from **HotspotEth** to **bridge** to **HotspotVLAN** VLAN.

Using the command => *interface list expand=disabled reverse=enabled* Listing 5.12 on the next page can be extracted. It shows that **HotSpotIP** is connected to **HotspotEth** and that **HotspotEth** is connected to **bridge**.

```
=> interface list expand=disabled reverse=enabled
Name         Type     State        Use    LL Interfaces
----         ----     -----        ---    -------------
[...]
HotspotEth   eth      connected    1      bridge
[...]
HotSpotIP    ip       connected    0      HotspotEth
[...]
```

Listing 5.12: Using the interface command described in Section 3.2 the connection from **HotSpotIP** IP interface to **bridge** is shown.

5.1.4 Set up an IPv4 Subnet

HotSpotIP is now defined as an IP interface, but has no IP subnet defined. From Listing 5.13 it can be seen that the private network (**LocalNetwork**) uses the IPv4 subnet `10.0.0.1/24`. It is decided to use `192.168.13.32/28` as the IP subnet for **HotSpotIP** with `192.168.13.33` as the gateway address[7]. `192.168.0.0/16` is defined by The Internet Assigned Numbers Authority (IANA) for use in private networks [IANA] and `192.168.13.32/28`[8] is therefore a reasonable choice, as only a little amount of hotspot users should be connected (more on this in Section 5.1.6). Listing 5.13 also shows that **LocalNetwork** has a IPv6 subnet defined. IPv6 was disabled for **HotSpotIP** in the configuration in Listing 5.9, so no IPv6 subnet has to be defined.

```
=> ip iplist
Flags legend: [P]referred   primar[Y]   [R]oute       [H]ost route
              d[E]precated  [I]nvalid   [T]entative  d[U]plicated
              [A]nycast     auto[C]onf  [D]ynamic     [O]perational
Prefix            Interface     Type       Flags          Remote IP
------            ---------     ----       -----          ---------
10.0.0.1/24       LocalNetwork  Ethernet   PYRH.......O
127.0.0.1/32      loop          Internal   ...H......DO

Prefix                          Interface      Type       Flags
------                          ---------      ----       -----
fe80::9e97:26ff:fea0:e6f4/64    LocalNetwork   Ethernet   ..RH......DO
::1/128                         loop           Internal   ..RH......DO
```

Listing 5.13: Default IP setup of IP interfaces. **LocalNetwork** has the subnet `10.0.0.1` with netmask: 24. *LocalHost* belongs to the IP interface **loop**. Both **LocalNetwork** and **loop** has IPv6 enabled.

Listing 5.14 on the following page shows the configuration of the IP subnet for **HotSpotIP**.

[7] `192.168.13.32` is the network identifier and `192.168.13.47` is the broadcast address of the subnet

[8] With netmask: 28 i.e. submask: `255.255.255.240`, 4 bits are available for modification which gives a range of 14 hosts.

```
1 => ip ipadd intf=HotSpotIP addr=192.168.13.33 netmask=28
2 => ip ipconfig addr=192.168.13.33 preferred=enabled primary=enabled
```

Listing 5.14: Set up an IP subnet (`192.168.13.32/28` and gateway: `192.168.13.33`) for the **HotSpotIP** IP interface.

Line 1: Add the subnet `192.168.13.32/28` for **HotSpotIP** which has the gateway address `192.168.13.33`.

Line 2: Configure the gateway of the subnet.

preferred: Means that this IP is the preferred address for that subnet.
primary: Means that this IP is the primary address for that subnet.

Listing 5.15 shows that the **HotSpotIP** interface now has IP subnet: `192.168.13.32/28` and gateway: `192.168.13.33`.

```
=> ip iplist
Flags legend: [P]referred   primar[Y]    [R]oute      [H]ost route
              d[E]precated  [I]nvalid    [T]entative  d[U]plicated
              [A]nycast     auto[C]onf   [D]ynamic    [O]perational
Prefix            Interface    Type       Flags         Remote IP
------            ---------    ----       -----         ---------
192.169.13.33/28  HotSpotIP    Ethernet   PYRH.......O
10.0.0.1/24       LocalNetwork Ethernet   PYRH.......O
127.0.0.1/32      loop         Internal   ...H......DO

Prefix                         Interface     Type      Flags
------                         ---------     ----      -----
fe80::9e97:26ff:fea0:e6f4/64   LocalNetwork  Ethernet  ..RH......DO
::1/128                        loop          Internal  ..RH......DO
```

Listing 5.15: **HotSpotIP** now has the IPv4 subnet `192.168.13.32/28` with gateway `192.168.13.33`. It can be seen that **HotSpotIP** does not have a IPv6 subnet, because IPv6 was disabled in Listing 5.9 line 5. *The list shows the Prefix as gateway with netmask and not subnet with netmask.*

Network Address Translation (NAT)

As seen in Listing 5.16 NAT is *disabled* for **HotSpotIP**. As **HotSpotIP** has its own IP subnet, NAT should stay in this mode. The fact that NAT is *disabled* for **HotSpotIP** does not influence the public IP address as all traffic going external will go through the interfaces **internet_pppoa** or **internet_pppoe**, and it can be seen that NAT is *enabled* for both interfaces. As NAT is already *disabled* for **HotSpotIP** no configuration should be made.

```
=> nat iflist
Interface        NAT
---------        ---
loop             disabled
internet_pppoa   enabled
```

```
internet_pppoe  enabled
ip_voip         disabled
LocalNetwork    transparent
ip_wan_voip     disabled
HotSpotIP       disabled
```

Listing 5.16: The list shows the *NAT* settings of the router. NAT is currently *disabled* for **HotSpotIP**.

5.1.5 Enable DNS Server

The function of DNS was explained in Section 3.3.2. A connection to the local *DNS server* is opened for the *dmz*-group, which was defined during the configuration of the **HotSpotIP** IP interface in Listing 5.9. The DNS server will serve as a local DNS (cache) server, which will forward DNS lookups (if not cached), for hotspot users, to the Resolving Name Server located at Telenor. The configuration can be seen in Listing 5.17. The MAG has one predefined DNS server (*DNS-S*) and it is not possible to create a new one, which means that hotspot users will use the same DNS server as private users. Using one DNS server for the private network and the hotspot network introduces a problem that will be further investigated in Section 6.6. The problem arises when the DNS server caches hostnames and it then becomes possible to resolve hostnames between the two networks using a reverse DNS lookups.

```
=> service system ifadd name=DNS-S group=dmz
```

Listing 5.17: Enable a local DNS server for the group *dmz* i.e. **HotSpotIP**.

5.1.6 Enable DHCP

The **HotSpotIP** IP interface should have a DHCP server for the IP subnet defined in Section 5.1.3. Section 3.3.3 went through the purpose of the DHCP server. Listing 5.18 shows the DHCP server for **LocalNetwork**, which has the gateway address: `10.0.0.1` with netmask: `24` and thereby has the allocated IP range: `10.0.0.[2-254]`. It is decided that the DHCP server for **HotSpotIP** will only be able to have 8 clients connected at a time. This is to prevent overloading of the network which could influence overall performance for both private users and hotspot users. The **HotSpotIP** subnet will have the IP range `192.168.13.[37-44]`.

```
=> dhcp server pool list
Idx Pool          Address Range    Intf           Admin    Alloc    State
--------          -------------    ----           -----    -----    -----
0 LAN_private     10.0.0.[2-254]   LocalNetwork   UP       dynamic  static
```

Listing 5.18: Default DHCP pool of the router.

HotSpotIP will also need a DHCP relay agent, which is the routers localhost at IP `127.0.0.1`. In Listing 5.19 it can be seen that **LocalNetwork** already has a DHCP relay agent, but lacks one for **HotSpotIP**.

```
=> dhcp relay list
Name                        DHCP server   Interface     Giaddr
----                        -----------   ---------     ------
LocalNetwork_to_127.0.0.1   127.0.0.1     Localnetwork  10.0.0.1
```

Listing 5.19: The list shows that only **LocalNetwork** has a relaying gateway.

The configuration to add **HotSpotIP** in the *DHCP server pool* can be seen in Listing 5.20. The gateway address is `192.168.13.33` with netmask: `28`. The server can allocate 8 clients from `192.168.13.[37-44]`. Each client has a leasetime of 300s to prevent clients, who are not connected, taking up DHCP space. A client, who is not connected, could be a client that walks by the modem without actually using it. The short leasetime should not be a problem as clients will renew their leasetime when they have less than 50 percent lease time left. It only takes 2 packets to renew a lease as explained in Section 3.3.3.

Whenever a client wishes to get a DHCP lease they broadcast a *DHCPDISCOVER* message. The router collects the message at the *localhost* and looks to see if the message might be intended for someone in the DHCP server pool. The interface **HotSpotIP** will therefore need to be known by the a relaying agent, which is the *localhost* at `127.0.0.1`.

```
1 => dhcp server pool add name=HotspotDHCP
2 => dhcp server pool config name=HotspotDHCP intf=HotSpotIP
      poolstart=192.168.13.37 poolend=192.168.13.44 netmask=28
      gateway=192.168.13.33 leasetime=300 localdns=enabled
3 => dhcp relay ifconfig intf=HotSpotIP relay=enabled
4 => dhcp relay add name=HotspotRelay
5 => dhcp relay modify name=HotspotRelay addr=127.0.0.1 intf=
      HotSpotIP giaddr=192.168.13.33
```

Listing 5.20: Enable DHCP and add a relaying gateway for the **HotSpotIP** IP interface.

Line 1: Add a new server pool with the name **HotspotDHCP**

Line 2: Configure the server pool to the **HotSpotIP** IP interface with the pool `192.168.13.[37-44]`, gateway: `192.168.13.33` and netmask: `28`. The leasetime is 5 minutes. The local DNS server is enabled such that hotspot users can use the local DNS server (*DNS-S*).

Line 3: Enable *relay* for **HotSpotIP**.

Line 4: Add the relay **HotspotRelay**.

Line 5: Configures the relay to the *localhost* address and connects it to **HotSpotIP** with gateway `192.168.13.33`.
When a hotspot client wishes to connect, it broadcasts the DHCP discover, the

> DHCP discover is forwarded by the relay agent to the DHCP server. The DHCP server answers with a DHCP offer for the client, which the client can accept (or deny) with a DHCP request. The last step of the process is the DHCP ACK by the server.

Listing 5.21 shows that **HotSpotIP** now has a DHCP server: **HotspotDHCP**. Listing 5.22 shows that the DHCP relay agent now knows the IP interface **HotSpotIP**.

```
=> dhcp server pool list
Idx Pool        Address Range       Intf          Admin  Alloc    State
--------        -------------       ----          -----  -----    -----
0 LAN_private  10.0.0.[2-254]      LocalNetwork  UP     dynamic  static
1 HotspotDHCP  192.168.13.[37-46]  HotSpotIP     UP     dynamic  static
```

Listing 5.21: **HotSpotIP** now has a DHCP pool alongside the pool of **LocalNetwork**.

```
=> dhcp relay list
Name                        DHCP server  Interface     Giaddr
----                        -----------  ---------     ------
LocalNetwork_to_127.0.0.1   127.0.0.1    Localnetwork  10.0.0.1
HotspotRelay                127.0.0.1    HotspotIP     192.168.13.33
```

Listing 5.22: Both **HotSpotIP** and **LocalNetwork** has a relaying gateway for the DHCP server at `127.0.0.1`.

5.1.7 Examination of The Hotspot Network Configuration

Listing 5.23 is the final approval for the configuration obtained from => *interface list expand=disabled reverse=enabled*. The list in Listing 5.23 has been modified, but the full list can be found in Appendix C Section C on page 207. The list shown, as explained in Section 3.2.3 can be used to make a diagram of the whole configuration of the modem. Comparing Listing 5.23 to Figure 4.1 the configuration is approved as: **wl_ssid1_local0** is connected to **HotspotBridge**, which is a member of **bridge**. **bridge** is connected to **HotspotEth**, which is connected to **HotSpotIP**.

```
=> interface list expand=disabled reverse=enabled
Name                    Type      State           Use  LL Interfaces
----                    ----      -----           ---  -------------
[...]
bridge                  eth       connected       3   eth_voip,
                                                      LocalNetwork,
                                                      HotspotEth
OBC                     bridge    connected       1   bridge
[...]
wlif1                   physical  connected       1   WLAN
[...]
WLAN                    bridge    connected       1   bridge
[...]
LocalNetwork            ip        connected       0
```

```
[...]
wl_ssid1_local0      physical  connected      1   HotspotBridge
HotspotBridge        bridge    connected      1   bridge
HotspotEth           eth       connected      1   HotSpotIP
HotSpotIP            ip        connected      0
```

Listing 5.23: Interfaces list on MAG when the configuration has been done. The list has been modified to only show interesting lines.

5.2 Configuration of Firewall Rules

At this point the **HotSpotIP** interface is fully functional, even though it still has no firewall rules. The firewall rules for hotspot users should be very strict such that they are safe from private users and vice versa. For this reason it is necessary to implement the following rules:

Allow **HStoWAN**: Allow traffic from hotspot users to the internet.

Connection **WANtoHS**⋆: Allow traffic from the internet to hotspot users.

Deny **HStoALL**: Deny all traffic from hotspot users to other groups (except *WAN*)

Deny **ALLtoHS**: Deny all traffic to hotspot users from other groups (except *WAN*).

⋆ Incoming traffic should only be allowed if a connection is first created by a hotspot user.

These rules are made such that hotspot users are allowed to get traffic to/from the Internet, but are not allowed anything else e.g. to interact with each other. An introduction to firewall rules is to be found in Section 3.3.4. The firewall rules will be added to the *forward_level_Standard*, because it is only permitted to allow incoming traffic if a connection is first established by the host. The default firewall rules cannot be shown directly in the report, but can be found in Appendix D.1 on page 210.

In Section 3.3.4 it is explained that the chain: *forward_level_Standard* blocks all incoming traffic. The rule **WANtoHS** should therefore not be implemented, as incoming traffic is (indirectly) blocked by default. Traffic will still be able to go to hotspot users if they are creating the connection. Listing 5.24 shows firewall rules that are implemented.

```
=> firewall rule add chain=forward_level_Standard index=1 name=
    HStoWAN srcintf=dmz dstintf=wan state=enabled action=accept
=> firewall rule add chain=forward_level_Standard index=2 name=
    HStoALL srcintf=dmz state=enabled action=deny
=> firewall rule add chain=forward_level_Standard index=3 name=
    ALLtoHS dstintf=dmz state=enabled action=deny
```

Listing 5.24: Firewall rules to isolate hotspot users from everything but WAN.

The added firewall rules can be seen in Listing 5.25 and found in the Appendix at D.2 on page 212. Rules are read in sequential order starting from first index, and endning at first match. Meaning that: The firewall goes through the rules from index one and does not care about conflicts.

```
=> firewall rule list
Rules (flags: C=Constant, D=Dynamic, E=Enable, L=Log)
=====
Chain                    Nr. Flags  Rule Name  Action  Conditions
-----                    --- -----  ---------  ------  ----------
[...]
forward_level_Standard   1   C E    HStoWAN    accept  dmz.* > wan.*
                         2   C E    HStoALL    deny    dmz.* > *.*
                         3   C E    ALLtoHS    deny    *.* > dmz.*
                         4   C E    FromLAN    accept  lan.* > *.*
[...]
```

Listing 5.25: Firewall rules are now created for hotspot users in *group = dmz*. Rules having lowest index number has highest priority.

5.2.1 Block Access to Services

When firewall rules are implemented for interface to interface traffic it is time to look for vulnerabilities to the MAG itself. These vulnerabilities could be in the *sink* packet flow as mentioned in Section 3.3.4. Looking into *chain=sink_system_service* in Listing D.1, in Appendix D.1 on page 210, it can be seen that many services are still open for hotspot users. Some open services are e.g: Access to the MAG GUI, Telnet, FTP, VOIP_SIP etc.

It is desirable to allow the least amount of services as possible to hotspot users, as every service can have unforeseen vulnerabilities. Access to the MAG GUI or the Telnet service will, e.g. allow hotspot users to change settings in the MAG if they somehow get the password. Three services should be allowed for the hotpsot network: *DHCP* is needed for the hotspot clients to get an IP address, *DNS* is needed to be able to resolve IP addresses from domain names (unless if they have defined another DNS server) and *ICMP* should be allowed such that it is possible to send error messages and routing traffic to the router. The following rules will be added and the implementation can be seeen in Listing 5.26:

Accept **dmzdns**: Accept *DNS* traffic from *group = dmz* to the hotspot gateway at `192.168.13.33`.

Accept **dmzdhcp**: Accept *DHCP* traffic from *group = dmz*.

Accept **dmzicmp**: Accept *ICMP* traffic from *group = dmz* to the hotspot gateway at `192.168.13.33`.

Deny **dmzall**: Deny traffic from *group = dmz* to everything except *DNS* and *DHCP*

Deny **HSgateway**: Deny traffic from *group* = *lan* to the hotspot gateway.

```
=> firewall rule add chain=sink_system_service index=1 name=HSgateway srcintf=lan dstip=192.168.13.33 log=disabled state=enabled action=deny
=> firewall rule add chain=sink_system_service index=2 name=dmzicmp srcintf=dmz dstip=192.168.13.33 serv=icmp log=disabled state=enabled action=accept
=> firewall rule add chain=sink_system_service index=3 name=dmzdns srcintf=dmz dstip=192.168.13.33 serv=dns log=disabled state=enabled action=accept
=> firewall rule add chain=sink_system_service index=4 name=dmzdhcp srcintf=dmz serv=dhcp log=disabled state=enabled action=accept
=> firewall rule add chain=sink_system_service index=5 name=dmzall srcintf=dmz log=disabled state=enabled action=deny
```

Listing 5.26: Make firewall such that hotspot users can not access the services they should not use.

The added rules and be seen in Listing D.2, in Appendix D.2 on page 212 in *chain=sink_system_service*. A snapshot is shown in Listing 5.27. The expressions *icmp, dns, dhcp* can be extracted using => *expr list* and can be seen in Appendix D.3 on page 215.

```
=> firewall rule list
Rules (flags: C=Constant, D=Dynamic, E=Enable, L=Log)
=====
Chain   Nr. Flags  Rule Name Action  Conditions
-----   --- -----  --------- ------  ----------
[...]
sink_system_service
        1   C E    HSgateway deny    lan.* > *.192.168.13.33
        1   C E    dmzicmp   accept  icmp dmz.* > *.192.168.13.33
        2   C E    dmzdhcp   accept  dhcp dmz.* > *.*
        3   C E    dmzdns    accept  dns  dmz.* > *.192.168.13.33
        4   C E    dmzall    deny    dmz.* > *.*
        [...]
```

Listing 5.27: The added firewall rules will block access to anything but *DNS*, *DHCP* and *ICMP* for hotspot users, belonging to the group *dmz*.

Now that the configuration of the MAG is finished it is time to test the setup. It is at this point assumed that hotspot hosts can get an IP address, connect and use the Internet. The next Chapter will look into the security of the setup and use different methods to scan the network for vulnerabilities.

CHAPTER

SIX

TEST

This Chapter will test the security of the newly implemented configuration of the MAG. The idea is to use *Nmap*, short for Network Mapper, to do different network scans to check for vulnerabilities that might decrease the level of security by the implementation of the hotspot network. It is chosen to use Nmap as it is the most popular tool for the type of scans this project will consider. This Chapter will consist of an introduction to Nmap followed by different kinds of tests:

- Ping scan
- TCP port scan
- UDP port scan
- ICMP scan
- Network Basic Input Output System (NetBIOS) scan - *It has been considered to do a NetBIOS scan, but the new verison of NetBIOS uses the TCP and UDP protocols, so open services will be seen in those scans.*

6.1 Network Mapper - Nmap

Nmap is a tool for "*network discovery and security auditing*" [Nmap [2015b]]. It is used to find possible penetration vulnerabilities in both small and large scale networks by using different types of network scans. Such scans can be used by intruders to get access to certain services of a system e.g. to find an open TCP port and thereby introduce malicious software to an Operating System (OS). Nmap can, however, also be used to find such vulnerabilities before a system becomes operational, such that network administrators can ensure network security by closing services or adding firewalls. Some functions of Nmap worth mentioning are:

- Discover hosts using ping.
- Determine running OS on a host.
- Scan for open TCP and UDP ports (services).
- Scan and discover multiple hosts in large scale networks.

During a scan, Nmap creates a table with: *port number* and *protocol*, *service name*, and *state*. An example is port/protocol: `53/TCP`, service: *domain*, state: *open*. It means that port 53 is open for the TCP protocol. Port 53 is, in this example, used by a DNS-server. Nmap considers ports in one of the following states [Nmap [2015d]]:

- **Open** ports accepts a TCP or UDP connection actively. Open ports are what penetration testers are typically looking for as this is where the most harm can be done to a system.
- **Closed** ports are reachable, but no application is listening. It does however show that a host is accessible on that IP.
- **Filtered** ports are ports that give no response (or some `ICMP unreachable` errors). It could mean that there is no host or that something is blocking probes e.g. a firewall.
- **Unfiltered** ports are reachable, but it is not possible to determine if the port is **open** or **closed**. Other scan types can resolve this issue.
- **Open|Filtered** means that Nmap cannot determine if a port is open or filtered. This happens when a port does not return a response e.g. UDP scans.
- **Closed|Filtered** means that Nmap cannot determine if a port is closed or filtered.

The next Section will look into the different kind of Nmap scans that will be done.

6.2 Nmap Scans

Nmap scans will be done from a computer running Kali Linux, which is based on Debian. The scans are started from Linux terminal and follows this setup: *nmap [<Scan Type> ...] [<Options>] <target specification>* [Nmap [2015c]]. A *scan type* is e.g. a TCP scan, *options* could be a specification of which ports to scan and *target specification* is the IP address to scan [Nmap [2015c]]. The scans that will be done in this project will each be explained.

6.2.1 ICMP Echo (ping) Scan

Why A ping scan is done, because it is interesting to see which hosts can be discovered.

How Ping uses the Internet Control Message Protocol (ICMP) protocol to test network connectivity. It sends an ICMP echo request and waits for an answer. It is commonly used to check network connectivity, to measure Round Trip Time (RTT) and count packet losses [Cisco [2006]].

With *nmap -sn <target specification>*
This scan will simply ping IP addresses to look for echoes from online hosts.

- **-sn**: Performs no port scanning. It discovers reachable hosts using a ping sweep (*ICMP echo request*). **-sn** also sends a *TCP SYN* to port `443`, a *TCP ACK* to port `80`, an *ICMP timestamp request* as well as an *ARP request*[Nmap [2015c]].

6.2.2 TCP SYN Scan

Why A TCP scan is done to test the state of TCP ports. Sometimes it is possible to connect to a TCP port even though it looks like a host is down. This test can be useful to discover hosts as well, or worse, find open ports.

How The TCP connect scan sends a connect request to a port and waits for a response. From the response (or missing response) Nmap can determine the state of the port.

With *nmap -Pn -sS -p 1- <Target specification>*
This scan will skip host discovery. It scans all TCP ports to determine their state.

- **-Pn**: Skip host discovery.
- **-sS**: TCP SYN scan sends a `SYN` packet, for a specific port, to the target. Using only the `SYN` packet, the connection is never fully established and is therefore considered relatively stealthy. From the response Nmap can determine the state of the port:

  closed: `RST`
  open: `SYN/ACK`
  filtered: `No response` or `ICMP unreacable` errors

  Another option is to use the TCP connect scan (**-sT**). The **-sT** scan does not use the usual (`SYN`) connect, but asks the operating system to use a high-level *connect* system call to establish a connection to the target. It is chosen to only use the TCP SYN scan for this project, because this scan is usually used. TCP connect scan is only used for specific purposes when the TCP SYN scan is not available[Nmap [2015c]].
- **-p**: Is to define a specific port range to scan. **1-** scans all ports. Usually Nmap will only scan the most common ports.

6.2.3 UDP Scan

Why A UDP scan is done to test the state of UDP ports. Sometimes it is possible to send packages to a UDP port even though it looks like a host is down. This test can be useful to discover hosts, or worse, find open UDP ports.

How The UDP scan sends a packet to a port and waits for a response. From the response (or missing response) Nmap can determine the state of the port.

With *nmap -Pn -sU -reason -p 1- <Target specification>*
This scan will skip host discovery. It scans all UDP ports to determine their state.

- **-Pn**: Skip host discovery.
- **-sU**: The UDP scan sends an empty packet (protocol-specific packet in some cases) to a port. From the response the state of the port can be determined:

  closed: `ICMP unreachable` errors
  open: response from port
  open|filtered: no response from port

  UDP scans are typically very slow compared to TCP. This is because it is more difficult to determine the state of a port when there is no response. The packet could be lost (retransmission) or the packet could simply have been sent to an open or filtered port. Due to that and because some machines have restrictions to how quickly `ICMP unreachable` errors can be sent (one per second) a UDP scan for all ports (`1-65536`) can take more than 18 hours[Nmap [2015c]].
- **-reason**: Gives an deeper explanation of the port state. *Scans with **-reason** will not be used.*
- **-p**: Is to define a specific port range to scan. **1-** scans all ports. Usually Nmap will only scan the most common ports.

6.2.4 ICMP Scan

Why ICMP messages are used for error handling and routing. Using different ICMP messages it might be possible to harm a system.

How Different ICMP messages will be sent to different types of systems. Looking into the replies it might be possible to find some kind of vulnerability.

- It is chosen to only do the ICMP echo scan due to time restrictions for the project. It is unknown which results would be discovered and their influence.

The next Section will go through the test setup for the scans.

6.3 Test Setup

To test the network a setup is proposed as seen in Figure 6.1. The test includes the following connected devices for each network: one Windows 7 computer, two Raspberry Pis and the gateway for each network as well as two disconnected, but leased hosts, which is a Windows 7 computer and a Raspberry Pi computer. A wired Raspberry Pi is connected to an Ethernet port for the private network. A computer running Nmap will be connected to the network from which the test is done from.

The public IP of the modem is also included in the scan as threats from the outside is also of concern for both networks. Using this setup both wired and wireless devices are scanned as well as two different OSs: Windows and Raspbian (Linux Debian). It is known that some devices are not included in the test[1], but it is not possible to test every device, running every operating systems, running every service possible. It is decided to only do the Nmap scans from a computer running Kali Linux (Debian) even though scans from different OSs could potentially see different things e.g. the old NetBIOS between Windows computers.

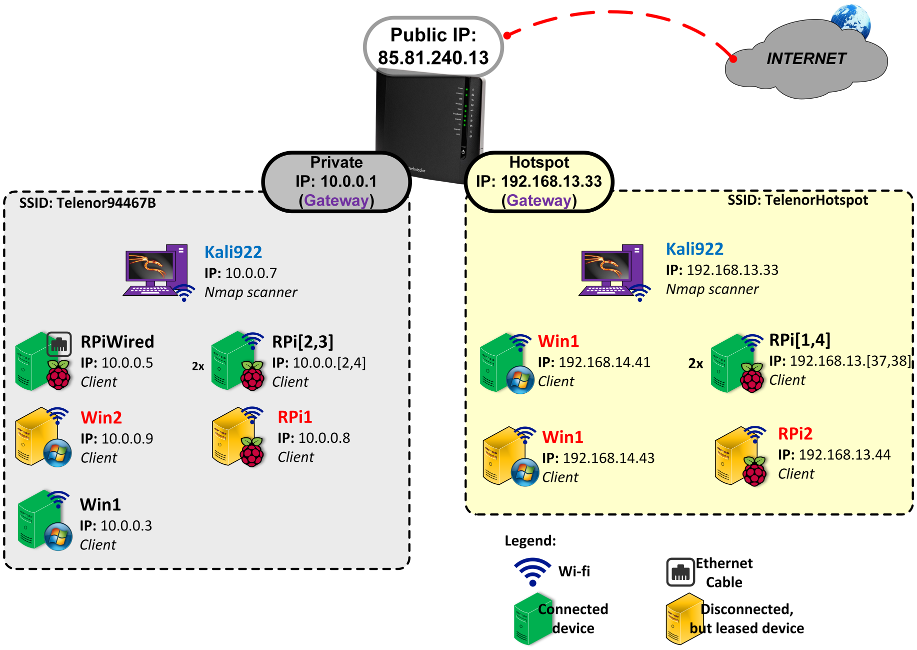


Figure 6.1: Network test setup for network integrity scans.

The list in Listing 6.1 is taken from the DHCP server of the MAG, using the CLI command: => *hostmgr list*. It is seen that the list approves the test setup proposed which both connected and leased devices.

[1] Android phones, Windows phones, Iphones and other Mac devices, Solaris OS, different versions of Windows, different distributions of Linux and so on.

```
=> hostmgr list
IPv4-address [...] Flags IP.Intf      Hw Intf         Hostname
------------       ----- ----         -------         --------
10.0.0.1           CT    LocalNetwork -               localhost
10.0.0.2           CD    LocalNetwork wlif1           RPi2
10.0.0.3           CD    LocalNetwork wlif1           Win1
10.0.0.4           CD    LocalNetwork wlif1           RPi3
10.0.0.5           CD    LocalNetwork ethif4          RPiWired
10.0.0.7           CD    LocalNetwork wlif1           kali922
10.0.0.8           D     LocalNetwork wlif1           RPi1
10.0.0.9           D     LocalNetwork wlif1           Win2
192.168.13.37      CD    HotSpotIP    wl_ssid1_local0 RPi1
192.168.13.38      CD    HotSpotIP    wl_ssid1_local0 RPi4
192.168.13.41      CD    HotSpotIP    wl_ssid1_local0 Win2
192.168.13.42      D     HotSpotIP    wl_ssid1_local0 kali922
192.168.13.43      D     HotSpotIP    wl_ssid1_local0 Win1
192.168.13.44      D     HotSpotIP    wl_ssid1_local0 RPi2
```

Listing 6.1: List of connected devices. Deleted columns: *MAC-address*, *IPv6-address*, and *Type*. Output changed a bit to make space for *Flags*. Flags: *C*=connected, *D*=Leased, *T*=modem.

The following list is based on the list from Listing 6.1 and shows the name, IP, OS, network and service(s) for each device. red means leased device, and blue means that the device can switch between the hotspot network and the private network depending on the test.

- **Private network**

  Gateway (Linux - Technicolor specific): `10.0.0.1`
  RPi2* (Linux Debian, Raspbian): `10.0.0.2`
  Win1 (Windows 7): `10.0.0.3`
  RPi3* (Linux Debian, Raspbian): `10.0.0.4`
  RPiWired* (Linux Debian, Raspbian): `10.0.0.5`
  Kali922 (Linux Debian, Kali Linux): `10.0.0.7`
  RPi1 (Linux Debian, Raspbian): `10.0.0.8`
  Win2 (Windows 7): `10.0.0.9`

- **Hotspot network**

  Gateway (Linux): `192.168.13.33`
  RPi1* (Linux Debian, Raspbian): `192.168.13.37`
  RPi4* (Linux Debian, Raspbian): `192.168.13.38`
  Win2 (Windows 7): `192.168.13.41`
  Kali922 (Linux Debian, Kali Linux): `192.168.13.42`
  Win1 (Windows 7): `192.168.13.43`
  RPi2 (Linux Debian, Raspbian): `192.168.13.44`

- **Public IP** (Linux): `85.81.240.13`

* Port: `TCP/22` used for SSH.

To ensure that a test will include both connected, leased, and unleased IPs as well as the gateways and the public IP, the target for the scans will be: <*Target specification*> = { `10.0.0.1 10.0.0.2-20 192.168.13.33 192.168.13.37-46 85.81.240.13` } [2].

Now that the different scan methods have been described and the test setup explained the following section will explain the results observed through tests.

6.4 Network Scan Results

Nmap scans have been performed from both the private network and from the hotspot network. The scans follow the test setup recently described. The results from Nmap are too long to show directly in the report, but can be found in Appendix E on page 219. The Appendix quickly summarizes the test setup and the different scans followed by the scan results. TCP and UDP scans will be performed simultaneously. All scans performed from the private network are shown first followed by the scans from the hotspot network. The following list shows the scans performed as well the their location in the Appendix:

Section 6.4.1: ICMP echo (ping) scan

Private: Appendix E.0.1 on page 220
Hotspot: Appendix E.0.3 on page 224

Section 6.4.2: TCP port scan

Private$^\diamond$: Appendix E.0.2 on page 220
Hotspot$^\diamond$: Appendix E.0.4 on page 224

Section 6.4.3: UDP port scan

Private$^\diamond$: Appendix E.0.2 on page 220
Hotspot$^\diamond$: Appendix E.0.4 on page 224

$^\diamond$: TCP and UDP scans are done at once.

6.4.1 ICMP Echo (ping) Scan

In Sections E.0.1 on page 220 (private) and E.0.3 on page 224 (hotspot) the results of the ping scans can be seen. Table 6.1 summarizes the results obtained from the Nmap scan.

Private

From the table it can be seen that it is possible to ping, obtain host name and see the

[2] It is chosen not to include the whole `10.0.0.0/24` network, but only 19 leasable hosts (*10.0.0.2-20*) as it would simply take to much time to scan the whole network.

To / From	Private network	Hotspot
Private network	Ping, host name, MAC address	—
Private gateway	Ping, host name, MAC address	—
Hotspot	—	—
Hotspot gateway	Ping, host name	Ping, host name, MAC address
Public IP	Ping, host name	—

Table 6.1: Results from nmap ping scan.

MAC address of the private gateway (`10.0.0.1`) and private hosts (`10.0.0.2-20`) as desired. It is also possible to ping and get the host name of the public IP (`85.81.240.13`) and the hotspot gateway (`192.168.13.33`)[3]. It should not be a problem that it is possible to ping the hotspot gateway from private hosts as long they do not create a route to hotspot hosts. It is not possible to ping hotspot hosts (`192.168.13.37-46`), which is very important.

Hotspot

From the hotspot it is not possible to ping anything but the hotspot gateway which is as desired.

6.4.2 TCP port Scan

In Sections E.0.2 on page 220 (private) and E.0.4 on page 224 (hotspot) the results of the TCP port scans can be seen. During the scans it was discovered that:

To / From	Private network	Hotspot
Private network	TCP ports, host name, MAC address	host name
Private gateway	TCP ports, host name, MAC address	host name
Hotspot	—	—
Hotspot gateway	—	`53/tcp`, host name, MAC address
Public IP	TCP ports, host name	host name

Table 6.2: Results from nmap TCP (`SYN`) port scan. *TCP ports* means that multiple TCP ports are accessible.

Private

From the TCP `SYN` scan it can be seen that it is not possible to obtain host names or access TCP services of hotspot hosts (`192.168.13.37-46`) or the hotspot gateway (`192.13.33.33`). TCP ports are, as expected, accessible for hosts on the private network (`10.0.0.2-20`), the private gateway (`10.0.0.1`) and to the public IP (`85.81.240.13`). The access to the hotspot gateway is *filtered* by the added firewall rule in *chain=sink_system_service* that denies traffic from group `lan` to the hotspot gateway in Section 5.2.1 on page 63.

[3] It was observed that the the linux command *ping 192.168.33.33* did not get a reply.

Hotspot

Hotspot hosts are unfortunately able to obtain host names of private hosts and the private gateway. This is an issue as host anonymity is compromised, which might be a problem. The fact that the problem originated only because one DNS server is available, is to be discussed in Section 6.6 on page 76. It is not desired to see this behaviour, but as long as the MAG has only one DNS server, there is no direct solution to the problem. Hotspot hosts can only access `53/tcp` on the hotspot gateway, which is accepted as this port is used to access the DNS service.

6.4.3 UDP port Scan

In Sections E.0.2 on page 220 (private) and E.0.4 on page 224 (hotspot) the results of the UDP port scans can be seen. Table 6.3 summarizes the results obtained. It should be noted that *open|filtered* means that a port state cannot be determined by the UDP scan. *open|filtered* ports are not included in the Table. During the scans it was discovered that:

To / From	Private network	Hotspot
Private network	`123/udp`, `137/udp`, `5353/udp`, host name, MAC address	host name
Private gateway	`53/udp`, host name, MAC address	host name
Hotspot	—	—
Hotspot gateway	—	`53/udp`, host name, MAC address
Public IP	`53/udp`, host name	host name

Table 6.3: Results from nmap UDP port scan. *UDP ports* means that open UDP ports are accessible. *open|filtered* ports are not shown.

Private

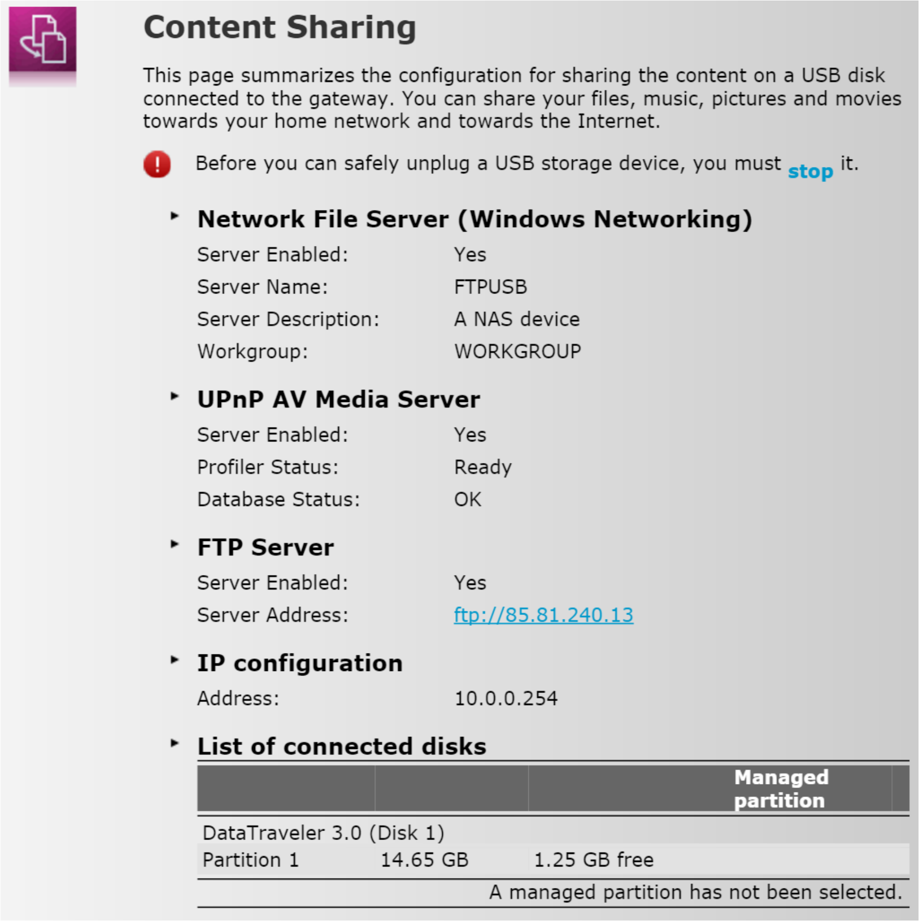
Open ports, host name, MAC addresses can be accessed/obtained for private hosts, the private gateway, and the public IP as expected and as seen for the TCP `SYN` scan. Hotspot hosts and the hotspot gateway are hidden for private hosts, which is the ideal scenario.

Hotspot

As for the TCP `SYN` scan from the hotspot network, hotspot hosts can also get access to the host names of private hosts (see Section 6.6 on page 76). It is not desirable, but as long as the MAG only has one DNS server there is no solution to the problem. Hotspot hosts can only access `53/udp`, which is accepted, as this port is used to access the DNS service.

6.5 Scan of Network Attached Storage Device

To finalize the scans, a storage device is connected to the USB port on the back of the MAG; It is now a Network File Server (NFS) device. Using the GUI of the MAG a File Transfer Protocol (FTP) server is started, which is open to both the private and public sides of the network. The intention of this scan is to investigate if and how hotspot clients can access the storage device. It is chosen to set up the FTP server from the GUI as it is how a potential user would set up the NFS device. The setup is seen in Figure 6.2.



Content Sharing

This page summarizes the configuration for sharing the content on a USB disk connected to the gateway. You can share your files, music, pictures and movies towards your home network and towards the Internet.

Before you can safely unplug a USB storage device, you must stop it.

Network File Server (Windows Networking)

Server Enabled:	Yes
Server Name:	FTPUSB
Server Description:	A NAS device
Workgroup:	WORKGROUP

UPnP AV Media Server

Server Enabled:	Yes
Profiler Status:	Ready
Database Status:	OK

FTP Server

Server Enabled:	Yes
Server Address:	ftp://85.81.240.13

IP configuration

Address:	10.0.0.254

List of connected disks

		Managed partition
DataTraveler 3.0 (Disk 1)		
Partition 1	14.65 GB	1.25 GB free

A managed partition has not been selected.

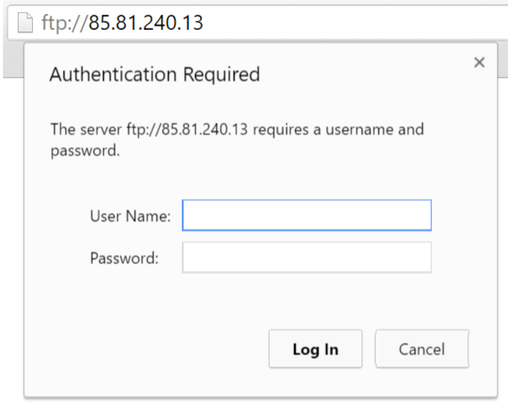
Figure 6.2: The Figure shows how the FTP setup looks for a potential user in the GUI.

It is observed that the NFS device can be accessed in two ways: 1. inside the private network at IP address `10.0.0.254` and 2. from the Internet at IP address `85.81.240.13`. It is expected that the FTP server does not introduce any direct vulnerabilities between clients in the two networks. Unless, of course, infected files are uploaded to the FTP server. It is expected that the FTP server opens a service at port 21, which is usually used for FTP services. To test this claim a TCP and UDP scan, following the same procedure as earlier, is performed on each of the IP addresses. The scan will be performed from the private network to see which ports are actually opened and a scan is performed from the hotspot network to test if and how the ports can be accessed. The scan result from the private network can be seen in Listing E.5 on page 227. In Listing E.5 interesting ports have been marked. These are the ports that were not present in the earlier scan and must have been opened by the NFS device. The scan result from the hotspot network is shown in Listing E.6 on page 228 also with interesting ports marked.

From the scans it is seen that:

- **Private Network:**
  - `10.0.0.254`: FTP services can be accessed at port `21/tcp`.
  - `85.81.240.13`: FTP services can be accessed at ports `21/tcp` and `2121/tcp`.
- **Hotspot Network:**
  - `10.0.0.254`: No FTP services available.
  - `85.81.240.13`: FTP services can be accessed at port `21/tcp`.

From the private network all FTP services should be open such that clients on the private network can access content on the NFS device. From the hotspot network it is expected, from earlier TCP and UDP scans, that all services in the private network are closed, which is why no services are open for hotspot clients at `10.0.0.254`. From the scan it is seen that hotspot clients are able to access port `21/tcp` at the public IP (`85.81.240.13`). At first it might look as a vulnerability, but further investigation reveals that an authentication password is still needed to really access the content of the FTP server. It is, in fact, discovered that everyone having the public IP of the modem are able to access the login screen of the FTP server at `ftp://85.81.240.13:21` in a browser (See Figure 6.3). As the FTP server is open to the public network and an authentication password is required, it is chosen to approve the setup as it is. It is important to mention that hotspot clients are able to access the FTP server and that the private user must set a difficult password. It should be noted that it is not possible to block access to the FTP server by changing firewall rules. This is due to the rule *ContentFTP* in *forward_level* (*index=2*) that allows *ftp* services from *wan* to go the *contentsharing_ip* in *lan*. When hotspot clients try to connect to the public IP (`85.81.240.13`) their traffic is routed as *wan* traffic and they are then allowed to access the FTP server. In *forward_level_Standard* it is possible to implement a rule that denies hotspot client to send *ftp* traffic to the *wan*, but this would apply to all FTP services on the Internet and are therefore not a possibility.



ftp://85.81.240.13

Authentication Required

The server ftp://85.81.240.13 requires a username and password.

User Name:

Password:

Figure 6.3: The login screen to the FTP server when the FTP server is accessed from the Internet.

6.6 Nmap & Reverse DNS Resolver

It was seen that hostnames were discovered during scans to/from all networks. The problem is quite interesting as a hostname cannot directly be used to do much harm, but it might be a problem with regards to host anonymity. Looking into the problem it found that Nmap uses *reverse DNS resolution* to discover host names[Benny [2011]].

Using the `in-addr.arpa` (inverse address) domain the reverse DNS resolver can translate an IP address into a hostname if there is a pointer record for that host [freesoft.org [2015]]. Reverse DNS lookup is most often used to authenticate hosts by translating an IP address into a hostname. This authentication is often done during email spam filtering or for system logging[The Texas Higher Education Network [2011]]. The process can be done as the MAG only has one DNS server and all leased hosts are stored with a pointer record, which points at a unique hostname (and a DNS suffix). To further prove the claim that Nmap uses Reverse DNS lookup, the following commands are issued [Nmap [2015a]]:

- **-n**: Denies the use of the reverse DNS resolver to discover hostnames.
- **-R**: Always use the reverse DNS resolver to discover hostnames.

In Listing 6.2 it shows that RPi1.lan[4] can be seen when Nmap uses the **-R** scan type and is hidden when using the **-n** scan type. Unfortunately the MAG can only have one internal DNS server and this issue cannot be solved unless a later upgrade allows the introduction of a second DNS server or if a set of rules can be introduced to the DNS server.

[4] *RPi1* is the hostname and *lan* is the DNS suffix.

```
root@kali922:~# nmap -Pn -R 192.168.13.37
Starting Nmap 6.49BETA5 { https://nmap.org } at 2015-12-11
    12:56 CET
Nmap scan report for RPi1.lan (192.168.13.37)
Host is up (0.0018s latency).
All 1000 scanned ports on RPi1.lan (192.168.13.37) are filtered

Nmap done: 1 IP addresses (1 host up) scanned in 5.38 seconds

root@kali922:~# nmap -Pn -n 192.168.13.37
Starting Nmap 6.49BETA5 { https://nmap.org } at 2015-12-11
    12:56 CET
Nmap scan report for 192.168.13.37
Host is up (0.0019s latency).
All 1000 scanned ports on 192.168.13.37 are filtered

Nmap done: 1 IP addresses (1 host up) scanned in 4.71 seconds
```

Listing 6.2: Difference when using reverse DNS resolver vs. denying the use of reverser DNS resolver. The scan is done from the local network to a hotspot host.

Listing 6.3 shows the devices known by the DNS server. It points out that the MAG only has one DNS server, which knows all hosts for both networks.

```
=> dns server host list
Address          Type   Hostname     TTL (s)   Creator
<local>            A    dsldevice    1200      undefined
<local>          AAAA   dsldevice    1200      undefined
10.0.0.4         * A    RPi3         0         DHCP_Server
10.0.0.254       * A    Linux        0         DHCP_Server
10.0.0.3         * A    Win1         0         DHCP_Server
10.0.0.2         * A    RPi2         0         DHCP_Server
10.0.0.4         * A    RPi3         0         DHCP_Server
10.0.0.5         * A    RPiWired     0         DHCP_Server
192.168.13.42    * A    kali922      0         DHCP_Server
192.168.13.38    * A    RPi4         0         DHCP_Server
192.168.13.41    * A    Win2         0         DHCP_Server
192.168.13.37    * A    RPi1         0         DHCP_Server
```

Listing 6.3: Hosts known to the DNS server. These hosts can be found using reverse DNS resolution.

6.7 Test Conclusions

This Section will summarize the test conclusions in order to make the results easily accessible to the reader. From the private network the following results were gathered:

- ✓ Private users can ping each other
- ✓ Private users can see each other's open services e.g. UDP and TCP ports.

- ✓ Private users can get hostnames and MAC addresses of each other.
- ✓ Private users can access the FTP server both in the private network and the public IP address.
- *X* Private users can get hostnames of hotspot users.

From the hotspot network the following results were gathered:

- ✓ Hotspot users cannot ping the private network.
- ✓ Hotspot users cannot ping each other.
- ✓ Hotspot users can only access the most essential services of the MAG i.e. ICMP, DNS server, and DHCP.
- *X* Hotspot users can get hostnames of both the private network and the hotspot network.
- ✓ Hotspot users cannot access the FTP server in the private network.
- *X* Hotspot users can access the FTP server on the public network, but still requires a matching password.

The main reason that access between users and access to certain services in the MAG is blocked, is because of the firewall rules implemented in Sections 5.2 and 5.2.1. The rules are designed to block access to hotspot users from the private network, and to block everything from the hotspot network except access to the Internet as well as the most essential services i.e. DNS, DHCP, and ICMP (for error handling) messages.

Another important implementation with regards to security, is the configuration of the SSID in Listing 5.2, where *apisolation=enabled* is the interesting command. By enabling *apisolation*, clients are isolated from each other within the hotspot network, and it is one of the main reasons why access between hotspot users is blocked, as messages between clients within a VLAN does not go through the router (firewall), but are routed in the bridge. It has not been entirely proven, but it is thought that this is the reason why hotspot users can not do reverse DNS resolving on other hotspot users.

The results have summed up the most important test results. The only real issue is that hostnames can be gathered between both networks. The next Chapter will conclude on Network Integrity.

CHAPTER

SEVEN

CONCLUSION

The configuration of the MAG has been done in order to ensure separation, with regards to security, between hotspot to hotspot users, hotspot users to the private network, and from the private network to hotspot users. To ensure this separation, layers 1 to 3 were considered, where layers 2 and 3 were the most important. The antenna was allocated a new port, which was routed to the new VLAN created for the hotspot network at layer 2. An Ethernet interface was created, which connected the newly created VLAN and a new IP interface for the hotspot network at layer 3. A DHCP server was configured for the hotspot IP interface and the hotspot network was connected to the local DNS server. Finally, firewall rules were created that should ensure strict seperation between the networks in the router. The basic idea of the rules was to deny all services and connections in the router for hotspot clients, except a connection to the Internet (WAN) and to the most essential services of the MAG.

The scans with different clients connected to the MAG was done in order to test if the separation between clients belonging to different networks fulfills the requirements. During the scans it was discovered that it was possible to block ICMP echo messages, block TCP and UDP connections and connect to the Internet as wished. It was, however, discovered that hotspot clients are able to resolve hostnames on the private network (but not from hotspot to hotspot) from the IP address using reverse DNS resolving. It does require the hotspot clients to guess different IP addresses, but it does not take much time to scan a whole subnetd. The problem is that only one DNS server is present in the MAG, which means that host anonymity in the private network is somewhat compromised. Hostnames cannot be used for anything specific to penetrate a system, but it makes it possible for hotspot clients to e.g. see if a smartphone is/has been (lease time) presented in the network, to check if someone is home or simply to see who lives at/owns the MAG. It is currently unknown why it is not possible to resolve other hotspot clients, it might be because of the option *apisolation=enabled* in line 2 of Listing 5.2, which means that clients connected to the SSID are isolated from each other. Another interesting point is that it is not possible for clients connected to the private network to resolve hostnames of clients connected to the hotspot network, even though it is possible for hotspot clients for resolve hosts in the private network. It might be because of the *apisolation=enabled* as well.

The following list is a repetition of the list shown in the requirements (Chapter 4) of the configuration of the hotspot network for the MAG. The list shows which requirements are approved and which requirements are not approved by the configuration and network tests:

Note: Requirements marked green are approved, red are disapproved, and orange are more or less approved.

Must have

1. Hotspot users will not have access to any devices in the private network.
2. Hotspot users will not have access to each other.
3. Hotspot users will not have access to services in MAG.
4. Private users will have access to each other.
5. Private users will not have access to the anything in the hotspot network.
6. Private users will be able to access services in MAG.
7. Hotspot users will have access to the Internet.
8. It will not be possible to access hotspot devices from the public Internet.
9. Access to the Internet for private users will not be changed.
10. Hotspot users will have the same accessibility to the public FTP server as anyone else.

Should have

11. Users in the hotspot network will not be able to see if a devices in the hotspot network are connected or not.
12. Users in the hotspot network will not be able to see if a devices in the private network are connected or not.
13. Users in the private network will be able to see if a devices in the private network are connected or not.
14. Users in the private network will not be able to see if devices in the hotspot network are connected or not.
15. Hotspot users should not be able to get information of devices that have been or are connected.
16. Private users should not be able to get information of devices that have been or are connected to the hotpsot network.
17. Hotspot users will use the best wireless security available in the MAG.

18. The DHCP should only distribute IPv4 addresses to hotspot users.

From the list it can be seen that most requirements are approved. It has been chosen to disapprove requirements 14., 15., and 16., because it is possible the see leased devices in the private network from the hotspot network using the reverse DNS resolver, and private users can see who is connected with the command: => *hostmgr list*. From the hotspot network it is however, not possible to see if leased devices are connected or not, which means that requirements 11. and 12. are approved. Because of the minor issue regarding hostnames it can be said that it introduces a vulnerability even though the most demanding requirements are approved. Hotspot clients are able to access the login screen of the FTP server at the public IP (`85.81.240.13`), but it has been chosen to accept the setup as they do not have direct access without authentication. It is up to the subscriber of the MAG to set a password that is not easy to guess. All devices connected to the Internet can access the FTP server on equal terms as hotspot clients. It is appropriate to note here that even though hotspot users can access the login screen of the FTP server on the public IP, they can not access the the Telnet service or the MAG GUI on either the public or the private IP address because firewalls in the *sink_service_system* chain denies it.

It should be noted that Telenor needs to prevent the xDSL subscriber from changing the configuration, which is a great threat to hotspot clients or the subscriber himself. The xDSL subscriber can easily change the setup to see traffic from hotspot users or introduce rouge DNS or DHCP server. Using a rouge DNS server it is possible to forward hotspot clients to harmful websites and with a rouge DHCP server it is possible to run code directly on a leasing device, every time it leases or renews its lease [Bull and Matthew [2015]].

sectionFurther work The next few sections will each comment on different aspects that might need further work in the project.

7.0.1 The Reverse DNS Lookup Issue

In Chapter 6 it was shown that it is possible to resolve hostnames in the private network from the hotspot network. See Section 6.6 for information on reverse DNS resolving. The issue might not be declared a direct vulnerability, but is still not an acceptable solution. The real problem is that the local DNS server (*DNS-S*) (Section 3.3.2) helps clients to resolve IP addresses by creating a recursive connection between a client and the local DNS server. A proposed solution to this problem is to create a second local DNS server, which is only for hotspot clients. With this, it might be possible to resolve hostnames of leased hotspot clients, but at least the private network is not compromised.

Another solution might be to not link hotspot clients to the local DNS server, but simple give them an IP address to a Telenor DNS server. In Section 3.3.3 it was mentioned that the DHCP server e.g. gives information about available DNS servers in the

DHCPOFFER message. This means that it is actually possible to push an IP address of a Telenor DNS server to connecting hotspot clients. Hotspot clients will thereby have to connect to each of the DNS servers themselves, but they are never linked to the local DNS server, which is a distinct advantage. This will of course change the configuration done in Chapter 5 and should be explained:

First of all it should be made sure that hotspot clients are never linked to the local DNS server. That configuration is done in Listing 5.17 on page 59, which only includes one CLI command that allows hotspot clients to use the local DNS server. The command should never be issued, because it lets hotspot clients resolve DNS addresses through the local DNS server.

In Listing 5.20 on page 60 the command in line 2 should include a Telenor DNS server. Telenor has two DNS servers available. The IP address of these servers are found by the CLI command: => *dns client dnslist* and gives the following IP addresses: `212.242.40.3` and `212.242.40.51`. The change is done by changing the line as seen in Listing 7.1, which gives a primary and secondary DNS server to hotspot clients and denies the use of the local DNS server.

```
=> dhcp server pool config name=HotspotDHCP intf=HotSpotIP
    poolstart=192.168.13.37 poolend=192.168.13.46 netmask=28
    gateway=192.168.13.33 leasetime=300 primdns=212.242.40.3 secdns
    =212.242.40.51 localdns=disabled
```

Listing 7.1: The Listing shows a modified version of Listing 5.20 on page 60, line 2. It now includes a primary and secondary DNS server for hotspot clients.

Finally the firewall rule *dmzdns* issued in Listing 5.26 on page 64, line 3 should be deleted as it allows the group *dmz* to connect to the local DNS server seen from a firewall perspective.

Due to time restrictions it is not possible to redo the tests of Chapter 6 and a quick test is simply performed to test if private clients are now hidden to hotspot clients. The Nmap command (*nmap -Pn -R x.x.x.x*) issued in Listing 6.2 on page 77 is used to force Nmap to do reverse DNS resolving. A Windows computer and a Raspberry Pi are connected to the private network while *kali922* is connected to the hotspot network. The results are as seen in Listing 7.2.

```
root@kali922:~# nmap -Pn -R 10.0.0.2

Starting Nmap 6.49BETA5 ( https://nmap.org ) at 2016-02-19 11:44
    CET
Nmap scan report for 10.0.0.2
Host is up (0.0039s latency).
All 1000 scanned ports on 10.0.0.2 are filtered

Nmap done: 1 IP address (1 host up) scanned in 4.28 seconds

root@kali922:~# nmap -Pn -R 10.0.0.3
```

```
Starting Nmap 6.49BETA5 ( https://nmap.org ) at 2016-02-19 11:44
    CET
Nmap scan report for 10.0.0.3
Host is up (0.011s latency).
All 1000 scanned ports on 10.0.0.3 are filtered

Nmap done: 1 IP address (1 host up) scanned in 4.74 seconds
```

Listing 7.2: A DNS scan is performed from a hotspot client that scans two clients connected to the private network. The scan is done while Nmap is forced to use the reverse DNS resolver (-*R*). The Listing shows how clients connected to the private network are now completely hidden for hotspot clients, by the fact it is now impossible to resolve hostnames.

The new scans show that requirement 15. is now approved. It is not possible to solve the issues with requirements 14. and 15. without blocking access to certain services in the MAG for the private user, or by changing the software to handle hotspot user information different.
Note: It has been chosen to only show the solution to the reverse DNS issue here, because of time restrictions. It would simply take too much time to redo the scans performed in Chapter 6.

7.0.2 Other Nmap Scans

During the Nmap scans it was chosen not to do a complete ICMP scan (due to time restrictions), but only try to use the ICMP echo message also known as a ping sweep scan. To finalize the scans performed it might be necessary to look into vulnerabilities due to ICMP messages. It might be that ICMP messages cannot be used to harm or penetrate a system, but research to confirm this needs to be done.

All scans performed were performed at layer 3. It might be interesting to look into possible vulnerabilities in other layers. All layers should be considered, but especially layer 2 might be of interest as it is the layer where VLANs and much configuration has been done during the setup. A known vulnerability could be a MAC flooding attack.

7.0.3 Creation of another Bridge Instance

In Chapter 3 the ability to add a second bridge instance for the hotspot network was considered (See Section 3.2.2 on page 32). The pros and cons of creating another bridge instance might in future work be considered in depth to draw the optimal conclusion. If the creation of another bridge instance is done, it could cause new vulnerabilities, which should also be looked into.

7.0.4 Configuration Change by The xDSL Subscriber

Nothing has been done in the configuration to deny the subscriber to change the configuration of the MAG. The configuration might be changed because a subscriber wishes to harm hotspot clients directly by e.g. collecting traffic to gather personal information. Another issue might be that a customer changes something in the configuration e.g. a firewall rule, without understanding the consequenses. It might, in worst case, let hotspot clients have access to applications or services that they should not have. Both of these are important considerations, but the latter might be the most important, because the introduction of a hotspot network might not be the choice of a subscriber, which could give Telenor problems when they introduce a vulnerability to a private network.

7.0.5 IPv6

During the configuration, in Listing 5.9 on page 55, IPv6 traffic was disabled because Telenor does not support IPv6 traffic yet. When/if Telenor introduces IPv6 traffic they should look into the possibility of creating an IPv6 hotspot network for hotspot clients. The implementation of IPv6 will open up new vulnerabilities, but it might also make parts of the configuration easier. Especially related to QoS and user identification.

The most important considerations of the Network Integrity part has now been considered. A combined list of the CLI command configurations of the MAG, shown in this Part, can be found in Appendix H Section H.1 on page 241. The Appendix have been modified a bit to save the amount of printed pages. The full list of unmodified configurations can be found in the *ListOfCLICommands.txt* file on the group server: `http://kom.aau.dk/group/16gr1022/`. The next part will look into how and if QoS can be ensured.

Part II

Quality of Service

CHAPTER

EIGHT

INTRODUCTION & ANALYSIS

Quality of Service (QoS) is the ability for an application to obtain the network service it requires for successful operation. Chapter 2 described the importance of prioritising the traffic of the paying subscriber, such that hotspot users will not influence the QoS of the private user(s). There are several ways to deal with QoS and the easiest way is to increase the bandwidth of the xDSL link, where the increase will only be available for hotspot users. A more innovative approach is to increase the utilisation of the already existing broadband connection. The idea is that hotspot users can use all bandwidth that is not utilized by the private user. This will provide a free service that requires no extra cost for the ISP. This approach however, introduces a QoS problem that involves using a service that the private user pays for. In order to solve the part of the problem formulation that states **"...no loss of QoS for the paying subscriber..."**, this chapter will describe the QoS capabilities of the MAG and analyse if and how different QoS parameters, used for prioritised queuing and scheduling, can be used to prioritise traffic of the private user and thus avoid loss of QoS.

It might seem trivial to apply QoS, but the complexity of the network and the amount of different links and layers as well as a lacking documentation of the MAG has made this task tedious. Large parts of this Chapter has required several experiments and educated guessing in order to effectively learn how different QoS capabilities in the MAG is designed and how they affect each other. Throughout the Chapter several issues will be introduced in terms of ensuring not only upload, but also download QoS. This involves not only the MAG, but also the ISP network devices as well as an analysis of the wireless medium and QoS limitations of the Wi-Fi protocol.

Section 8.1 will describe the overall QoS framework in the MAG and includes Ethernet QoS (bridge), wireless QoS (access point), and lastly IP QoS (router). Section 8.2 introduces an analysis of the possible bottlenecks in the network and ways to ensure that QoS is implemented in all critical points to ensure QoS for both upload and download traffic.

8.1 MAG QoS Framework

This section will describe the overall QoS framework of the MAG to provide an in-depth understanding of the different features supported. As described in Part I the MAG consists of several different elements, and each element has different QoS features. What is common for the router, bridge and AP is the classification of ingress traffic[1]. This is in order for that particular element to know how to treat the traffic in terms of queuing and scheduling. This means that in e.g. the case with upload traffic where the traffic comes from the bridge and arrives to the router, the router will classify the traffic *again*. The router can have different rules of classification than the bridge, so the queuing and scheduling priorities of the traffic can change throughout the MAG.

Classification of a frame, in the MAG, is done by assigning a priority, which will be referred to as the **internal class** of a frame. The internal class can have values ranging from *0-15* and higher numbers gives higher priority. This priority is used for the egress queuing in the router, switch or AP depending on the destination, to serve the higher priority packets first. But it is also used to ensure that low priority traffic will be denied from consuming memory resources in case of internal congestion or resource starvation. Telenor uses different network interfaces depending on the broadband connection resulting in different QoS queues. ATM interface is used only on ADSL links whereas Ethernet interface is used on VDSL and can be used on ADSL. The queues used for each interface are configured in the same way. It means that both interfaces uses the same configuration of queues and mapping of internal classes, meaning that the interface used will not have an effect as long as the queue configuration stay the same.

The framework uses the following elements, which operate independently of the other, to control the internal classes and provide QoS features throughout the MAG:

- Ethernet QoS in the Ethernet bridge (layer 2)
- Wireless QoS in the access point (layer 2)
- IP QoS in the router (layer 3)

A more thorough analysis of Ethernet QoS, Wireless QoS and IP QoS and their independent features for classification and queuing will now be made to provide a basis for understanding their relation and thus the overall picture of the QoS framework.

[1] Ingress and egress traffic refers to the traffic that enters or leaves the specific component in the MAG. Whereas upload or download traffic refers to the overall traffic stream throughout the MAG/network

8.1.1 Ethernet QoS

This Section takes basis in [*ConfigGuide_EthernetQoS.pdf*, Technicolor [2012b]]. Ethernet QoS provides QoS features of the Ethernet Switch, Bridge and the logical Ethernet interfaces. This part will focus on the QoS features of the bridge and logical Ethernet interfaces. The Ethernet switch is irrelevant for the scope of this project since it only affects the distribution of traffic between the physical switch ports. The logical Ethernet interfaces will by default add VLAN tags to each frame received. This VLAN tag has a three-bit field called the 802.1p user priority field. The field can be seen in figure 8.1. This is used to define a QoS parameter for the traffic that is visible to other layer 2 or 3 devices. The QoS parameter added is mapped from the internal class value of the frame received.

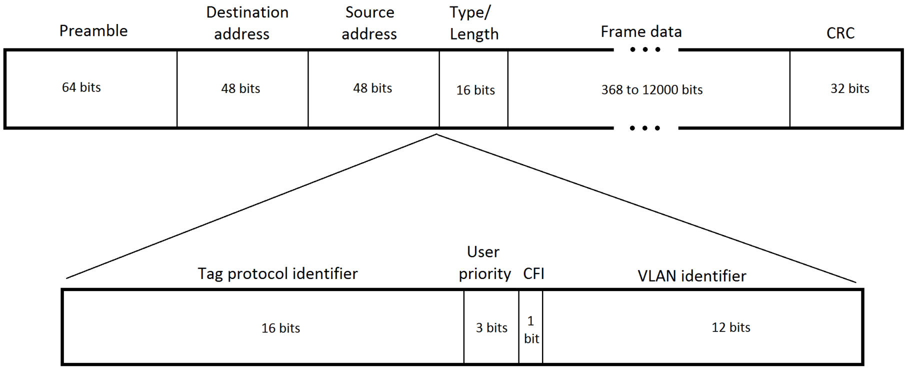


Figure 8.1: Illustrates a normal Ethernet frame being VLAN tagged.

The bridge has several QoS features and the most noteworthy for the scope of this project is classification of incoming frames. As mentioned earlier all traffic arriving at the bridge is classified with an internal class. The bridge can classify frames based on the following information:

802.1p user priority field

This classification method looks at the user priority field in the VLAN tag, and maps that priority to the internal class through a mapping table. This method is only available if the received frame has a VLAN tag, so this method will only be available in download stream cases. This is because packets arriving at the bridge in the upload stream have not yet been through a layer 2 device, like the logical Ethernet interface, that could have created a VLAN tag.

IP precedence field

The IP precedence field is a 3 bit field used for QoS in the IP Header. It ranges from 0-7 values where the higher the value the higher the priority for the IP packet. This field has largely been replaced with the DSCP field. This classification method requires an IP header.

DSCP field
The Differentiated Services Code Point field is a 6 bit field used for QoS in the IP Header. As mentioned, this has replaced the IP precedence field, however it is still backward compatible with the IP precedence field.

Port default priority
Each bridge port has a default priority which is used in cases where the previous information fields is unavailable. This means that all traffic arriving to a bridge port without any prior QoS parameters, is assigned with an internal class referring to the default priority of the specific port. This means that the port default priority is available in both download and upload streams, but used mostly in upload streams.

Medium priority
The medium priority is based on the medium the frame enters. The frame can enter either ATM or Ethernet interfaces. For Ethernet the priority will always be 4, for ATM the priority depends on the settings of the ATM QoS profile of that particular ATM interface. This classification method is available in both upload and download streams.

The first three are referred to as evaluated priorities as they rely on information from the received frame. Each of them is related to the following settings that can be assigned to bridge port, when doing Ethernet QoS:

Priority Config controls how to set the priority. It takes the following values as input:

disabled : uses the medium priority. By default it assigns internal class *2* to ATM interfaces and *4* to everything else.

overwrite : uses the evaluated priorities that relies on information from the received frame.

By default it is set to *disabled*.

IP Prec controls which of the evaluated priorities to use. It takes the following inputs:

precedence : uses the IP precedence value.

dscp : uses the DSCP value.

disabled : uses the 802.1p user priority value. If there are no 802.1p user priority it will set the priority to the default port priority.

Priority is the default port priority and takes the values *0-7*. Table 8.1 shows the mapping used in the bridge. The default port priority is mapped from the **VLAN & port** coloumn.

By understanding how the MAG maps the different values to internal classes is important to make sure the desired priorities is assigned correct. The next section will describe another important layer 2 device, namely the AP, and its QoS properties in order to apply QoS on the wireless medium.

Internal class	VLAN & port	IP precedence (ToS)	DSCP (DiffServ)
15	7	6, 7	CS6, CS7
14	6	5	CS5, EF
13	-	4	CS4, AF41
12	-	-	AF42, AF43
11	-	3	CS3, AF31
10	5	-	AF32, AF33
9	-	2	CS2, AF21
8	4	-	AF22, AF23
7	-	1	CS1, AF11
6	3	-	AF12, AF13
5	-	-	-
4	0	0	BE, CS0
3	-	-	-
2	2	-	-
1	-	-	-
0	1	-	-

Table 8.1: Bridge QoS mapping table

8.1.2 Wireless QoS

It is known, from the specification, that only one radio is available in the MAG. The radio is an electronic device that is used to either convert information into radio waves, that are propagated by the antenna, or to receive the waves from the antenna and transform them into a sequence of bits. Even though the MAG is capable of creating and maintaining up to 4 wireless networks, all of them will use the same radio which means they will use the same channel.

Wireless QoS in the MAG is based on the Wi-Fi Multimedia (WMM) technology. It has been introduced by the *Wi-Fi Alliance* and is based on the 802.11e standard [Society [2005]]. WMM uses a Enhanced Distributed Channel Access (EDCA) that gives, to specific applications, a higher probability to send the traffic. It does not guarantee any level of a throughput, it just gives different probabilities to access the medium so that, on average, traffic with higher priority will wait less time to send compared to traffic with lower priority. EDCA defines four types of queues called Access Categories (ACs):

AC_BK for *Background*: Low priority data that is not time-sensitive. For example data that needs high throughput to send large files.

AC_BE for *Best Effort*: Medium priority data that requires medium delay. Usually the queue for regular IP data with medium throughput.

AC_VI for *Video*: High priority data that is time-sensitive and requires minimum delay e.g. video.

AC_VO for *Voice*: Highest priority data that is very time-sensitive and requires minimum delay e.g. VoIP and streaming.

WMM defines the queues for two traffic directions - upstream and downstream:

Upstream is the data transfer from the host station to the Access Point (AP) and from the perspective of the MAG is defined as **ingress** traffic.

Downstream is the data transfer from the AP to the host station and from the perspective of the MAG is defined as **egress** traffic.

As the definition of wireless QoS for ingress and egress traffic is different from each other, each of them has to be described.

Egress Wireless QoS

In order to properly assign traffic to the right queue, the AP uses the internal class and VLAN priorities. Firstly, the internal class of the frame is mapped into a VLAN priority according to Table 8.2. Secondly, the VLAN priority is mapped into the appropriate WMM AC queue. For the wireless QoS to work properly for the egress data flow, WMM support is only required on the AP side.

Internal class	VLAN priority	Access Category (AC)
15	7	AC_VO (Voice)
14	6	AC_VO (Voice)
13	-	-
12	-	-
11	-	-
10	5	AC_VI (Video)
9	-	-
8	4	AC_VI (Video)
7	-	-
6	3	AC_BE (Best-effort)
5	-	-
4	0	AC_BE (Best-effort)
3	-	-
2	2	AC_BK (Background)
1	-	-
0	1	AC_BK (Background)

Table 8.2: Egress Wireless QoS mapping table

Wireless QoS for egress traffic is by default *enabled*, but Listing 8.1 shows how it is configured to use WMM.

```
wireless qos config mode=wmm
```

Listing 8.1: Enabling wireless QoS in the MAG.

Ingress Wireless QoS

The 802.11b/g/n protocols are not designed to guarantee upload QoS from a host station to the AP. The protocols are based on the Carrier Sense Multiple Access (CSMA) with Collision Avoidance (CSMA/CA) algorithm, where everyone is in fact competing against each other on the wireless medium. It means that ingress wireless QoS will instead depend on the host stations connected (wirelessly) to the MAG. The MAG does however also support WMM for ingress traffic, but it requires both the host station as well as a source application to support WMM. The source application have to be able to properly label traffic either with the layer 3 DSCP or layer 2 802.1p priority according to the traffic it generates. Moreover, the host station has to support and enable WMM in order to properly assign the traffic to the right AC queue based on the classification made by the source application. Unfortunately WMM is only suitable on host stations for basic traffic like voice or video, and *there is no way to force the host station to classify all traffic from one host station with a specific priority*, unless extra software is installed on the host station. It could e.g. be done in order to give all traffic coming from a host station low priority as it is needed for the purpose of this project.

8.1.3 IP QoS

IP QoS takes place in the router and provides layer 3 QoS features to the MAG. The general concept is the same as for Ethernet QoS and wireless QoS i.e. classification of ingress traffic and queuing of egress traffic, but with a few more features such as e.g. resource management. IP QoS can also be used to change the ToS field in the IPv4 header, which gives the possibility to identify traffic outside the MAG.

Resource management is handled through a series of operations which can be grouped into different components. A graphical overview of the main components in the upload data-path and their connections through the router is illustrated in Figure 8.2 that is based on [*ConfigGuide_IPQoS.pdf, Chapter 3*, Technicolor [2012b]].

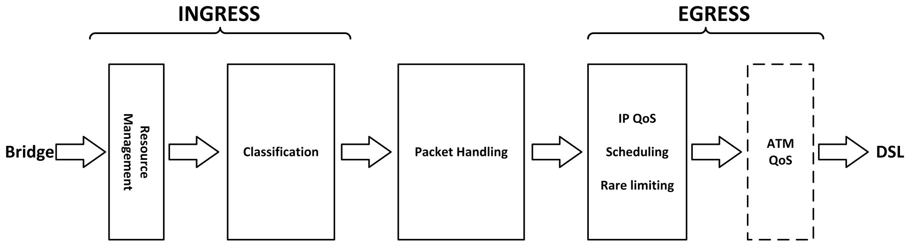


Figure 8.2: Graphical overview of the *main components* in the upstream datapath.

Each component in Figure 8.2 serves different purposes explained as follows:

Resource Management
The main purpose of this component is to reserve resources for higher priority traffic in the case of congestion or resource starvation. In these cases it will use the internal class priority assigned in layer 2 to deny low priority traffic from consuming memory resources.

Classification
This component classifies incoming data with a label. The label is only of internal significance and is used for QoS definitions which will be described later in section 8.1.3.

Packet Handling
In this component the IP packets goes through a series of processes like e.g. firewall, NAT and other routing operations.

IP QoS Queuing, Scheduling and Rate Limiting
This component controls the IP QoS queues and maps the internal classes to each queue. It provides several options such as rate limiting and scheduling.

ATM QoS
If the network infrastructure is ATM the modem will use ATM VP/VC queues instead of IP QoS queues.

For the download case it is expected that the router uses more or less the same components, but in a different order. The configuration guides does not describe the download case, but Figure 8.3 illustrates the expected download-path components and their connections.

IP QoS is applied to traffic in the router by classifying traffic with a label, according to a certain set of rules. It is done by having labels defined for a set of rules and then assigning a proper QoS parameter to each packet, if it complies to a certain rule. The QoS parameters that is assigned to a packet depend on the label of the rule it matches. The function of the QoS parameter is to decide the internal class for packets in the router according to a mapping table. An introduction to IP QoS is given by first

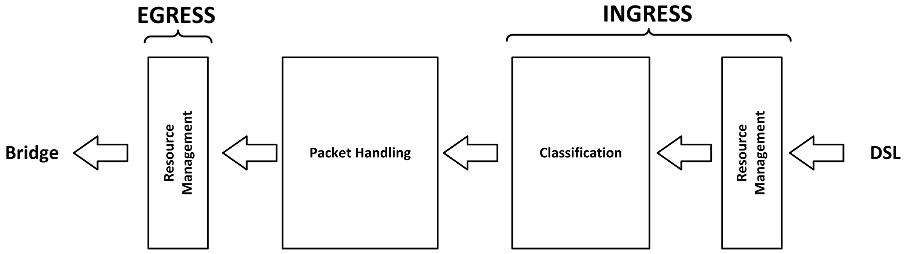


Figure 8.3: Graphical overview of the expected *main components* in the download datapath. The configuration guides does not describe the download case, but the Figure illustrates the expected download-path components and their connections.

analysing how the rules can be created and then looking into how the label can give QoS parameters to a packet. The IP QoS Section takes basis in [*ConfigGuide_IPQoS.pdf*, Technicolor [2012b]] as well as experience gained while working with the MAG.

IP QoS Rules

The function of the IP QoS rules is to classify traffic. Traffic can be classified in many ways and the rules created can, for example, depend on one or more of the following conditions:

Source interface e.g. *wan*, *lan*, or *dmz*

IP address(es) i.e. source or destination

Protocol e.g. TCP or UDP

Port(s) i.e. source or destination

QoS value i.e. DSCP, precedence, or internal class

and more...

As seen there are numerous ways to create specific rules/conditions to target specific traffic. Each rule defined for IP QoS is associated with a chain, an index, a name, a label, and a condition with each of the following properties:

chain: Chains are used for the ordering and grouping of rules. It is possible to assign a rule to a chain, a subchain within a chain, or a subchain of a subchain. *Some chains overrule or are overruled by other chains in the case of conflicting conditions.*

index: Each rule has an index within a chain. The indexes are read in sequential order and a label is assigned at first match.

name: The name is a description of a rule that does not have any function, but is simply used such that a rule can be identified.

label: A label is defined for each rule. The label includes the QoS parameters that will be assigned to a packet.

condition: The condition, combines only **srcintf**, **srcip**, **dstip**, and **serv**, and is the options to which a rule can comply. The condition entry has 4 entries and looks like: *serv 1.2 > 3.4* where:

> \>: Can be read as *going to*.
>
> *1*: To which source interface the rule applies.
>
> *2*: To which source IP address(es) the rule applies.
>
> *3*: To which destination interface the rule applies. *Note: It is not possible to apply conditions to the destination interface in IP QoS.*
>
> *4*: To which destination IP address(es) the rule applies.
>
> *serv*: To which services the rule applies to.

Each of the entries 1, 2, 3 or 4 can use the wildcard *****, which means ’*for all*’. Conditions can also be made for different services **serv** e.g. TCP, ICMP, or specific destination or source ports. The rule *ftp lan.* > *.192.168.1.1* will e.g. apply for FTP traffic (port: `21`) from *lan* going to IP `192.168.1.1`. The default expressions/services defined can be found in Listing D.3 on page 215 in Appendix D.

An example of how to add a rule is given in Listing 8.2. The Listing shows how a QoS rule is added to FTP traffic for the host belonging to the interface *lan* and sending traffic to IP address `192.168.1.1`. The rule is added to the chain *qos_user_labels* with index *3*. The rule has the name *FTPQoS* and is assigned the label *ftplabel*

```
=> label rule add chain=qos_user_labels index=3 name=FTPQoS srcintf
    =lan dstip=192.168.1.1 serv=ftp label=ftplabel
```

Listing 8.2: Example of how to add a QoS rule for FTP traffic for the host belonging to the interface *lan* and sending traffic to IP address `192.168.1.1`.

IP QoS Labels

Labels are defined in order to assign proper QoS parameters to a packet, when it complies to a rule. A label has various properties that are explained as:

name: The name of the label. The name is used to identify a label and is used when defining a IP QoS rule.

classification: Defines the method of the label. There are three different settings:

> *ignore*: Will not set or overwrite the existing packet class.

overwrite: Always overwrites the packet class to the one defined in the label.

increase: Will only overwrite the packet class if it is higher than the existing packet class.

defclass: Specifies the default internal class/QoS mapping of packets in the router. One of the three following options can be assigned:

0-15: A number between 0 and 15 assigns the internal class depending on the scheme seen in Table 8.3. Higher numbers results in higher priorities. Depending on the number the scheme will map the internal class to one of the five different internal queues, where *realtime* is the highest priority queue and *Best effort* is the lowest priority queue.

dscp: Uses the internal class depending on the *DiffServ DSCP* value defined in the **dscp** property below.

default: Gives the value 4 as the internal class.

ackclass: Specifies the internal class for acknowledgements. One of three options can be assigned:

0-15: A number between 0 and 15 can define the internal class as for **defclass** . It is also using the scheme seen in Table 8.3.

defclass: Uses the same parameter as defined in **defclass**.

prioritise: Gives a higher priority than defined in **defclass** (**defclass**+2).

bidirectional: Specifies if the returning stream will be assigned the same label. It uses the options *enabled* or *disabled*.

inheritance: Specifies if the label will be copied to child connections/streams. Child connections are connections that are setup by a parent connection. It uses the options *enabled* or *disabled*.

fapprio: Stands for Fair Access Policy (FAP) prioritisation and means that if several users are competing for throughput, the throughput will be shared fairly among the users. It uses the options *enabled* or *disabled*.

The following properties define if QoS parameters will be written to the Type of Service (ToS) parameter in the IPv4 header shown in Figure 8.4. The ToS parameter can be used to enable QoS on traffic in networks behind the MAG. The QoS properties are:

tosmarking: Determines if the ToS parameter in the IPv4 header will be changed or not[2]. It uses the options *enabled* or *disabled*.

tos: Specifies the ToS bits for the IPv4 header. It takes *values between 1 and 255* as valid input.

[2] The ToS byte in the IPv4 header can be used to define priorities for packets. The ToS byte is more or less outdated and does not have a specific function in the IPv4 header; except for QoS using the DSCP parameter [SAVVIUS [2016]].

precedence: Uses a *3 bit combination (8 values)* to define the ToS priority.

dscp: Is the most commonly used QoS definition as it is also backward compatible with both **tos** and **precedence**. It is written into the ToS parameter in the IPv4 header and takes the following options as input:

DSCP: Uses the scheme seen in the *DiffServ, DSCP* column in Table 8.3 with *Best effort* as the lowest priority and *CS7* as the highest priority.

0-63: Uses a number between 0 and 63 to map it into a DSCP parameter.

trace: Uses the options *enabled* or *disabled* to specify if IP tracing should be enabled or not.

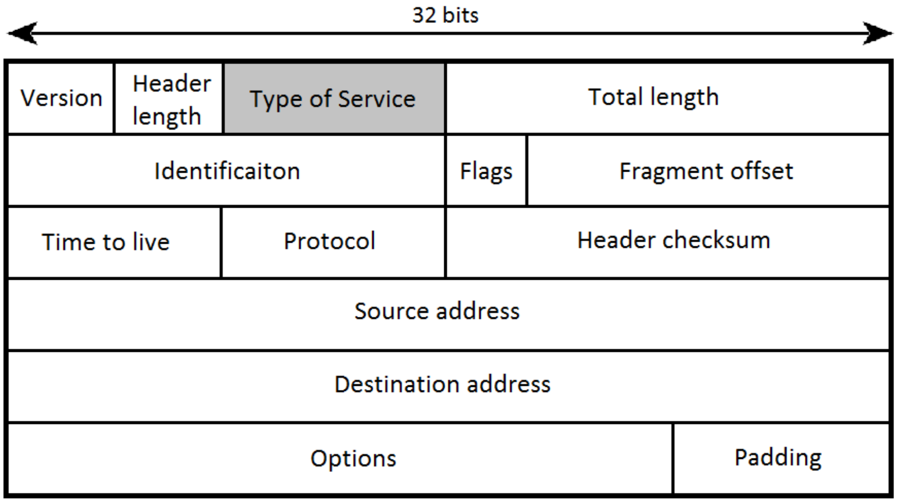


Figure 8.4: The IPv4 header, where the ToS field is marked in gray.

The properties for IP QoS labels has now been described. Listing 8.3 gives an example as how a label is created. The **name**/identifier of the label is *ftplabel*. The **classification** is set to *overwrite* meaning that **defclass** will change the internal class to *7* (resulting in the *WFQ1* queue as seen in Table 8.3) and the value of **ackclass** is *9* (**defclass**+2 for option *prioritise*). The label has **bidirectional** *enabled* such that the label will also be written to the returning stream, but **inheritance** is *disabled* such that the label is not written to child connections. **tosmarking** is *enabled* and the **dscp** value *cs1* is written in the IPv4 header. **fapprio** is *disabled* and the throughput will not be weighed fairly amongst competing users.

```
=> label add name=ftplabel
=> label modify name=ftplabel classification=overwrite defclass=7
     ackclass=prioritise bidirectional=enabled inheritance=disabled
     dscp=cs1 tosmarking=enabled fapprio=disabled
```

Listing 8.3: Example of how to add a QoS rule for FTP traffic for the host belonging to the interface *lan* and sending traffic to IP address `192.168.1.1`.

Input		Mapping	Output	
VLAN User Priority	DiffServ DSCP	Internal Class	Queue	Default label
7	CS6, CS7	15	5	Realtime
6	EF, CS5	14		
-	AF41, CS4	13	4	WFQ4
-	AF42, AF43	12		
-	AF31, CS3	11	3	WFQ3
5	AF32, AF33	10		
-	AF21, CS2	9	2	WFQ2
4	AF22, AF23	8		
-	AF11, CS1	7	1	WFQ1
3	AF12, AF13	6		
-	-	5	0	Best effort
0	CS0, Best effort	4		
-	-	3		
2	-	2		
-	-	1		
1	-	0		

Table 8.3: IP QoS scheme for the Technicolor TG788vn-v2 router [*Config-Guide_IPQoS.pdf*, page: 58 [Technicolor [2012b]]].

An introduction has now been given to the IP QoS framework of the MAG. The next section will analyse the network considered in the project in order to determine where bottlenecks might be located as it is an important consideration when doing QoS.

8.2 Network Overview and Bottleneck Analysis

This section will describe the basic network overview to analyse possible bottlenecks in the network observed. The bottleneck is the point in the network with the highest risk of network congestion. One of the methods to manage congestion is QoS, but if it is applied at the wrong point of the network it has no effect. In order to discover the right point to apply QoS a bottleneck analysis is performed. To illustrate the importance of a bottleneck analysis, a series of figures is shown in Figure 8.6. The Figures show a simplified case where a hotspot user and a private user is transmitting and receiving packets at the same time. This is to show how and where, in the network, QoS should be applied to prioritize traffic of the private user in the case of network congestion. The following description gives an explanation of Figures 8.6A-D:

Figure 8.6A illustrates two users (hotspot and private), that does download and upload of packets through two routers. The rates are not higher than what the different links can support so there is no network congestion and thus no packet drops.

Figure 8.6B illustrates the case where the link between the two routers is the bottleneck, which causes packets to queue up at each router. The network is congested and packets will eventually be dropped when the queue limit is reached. If no QoS is applied, packets from a private user are just as likely to be dropped, as packets from a hotspot user.

Figure 8.6C illustrates the same case as Figure 8.6B, but with QoS implemented at *Router 1* that prioritises packets from the private user. As seen, the router moves packets of the private user in front of the queue, delaying and eventually dropping packets from the hotspot user. As seen this does not solve congestion for the upload case, since QoS implemented in *Router 1* does not solve queuing issues in *Router 2*.

Figure 8.6D illustrates how QoS is implented in both *Router 1* and *Router 2*, that prioritises packets from the private user. It means that QoS now works in both the download and upload direction as QoS is on each side of the bottleneck.

Telenor has several different devices between the CPE and the Internet, where the most relevant for the scope of this project is the DSLAM and the BRAS. A simplified network overview is given in Appendix A on page 199 that explains the each device and their properties. Figure 8.5[3] shows these devices, that are the ones that can be configured in order to provide QoS.

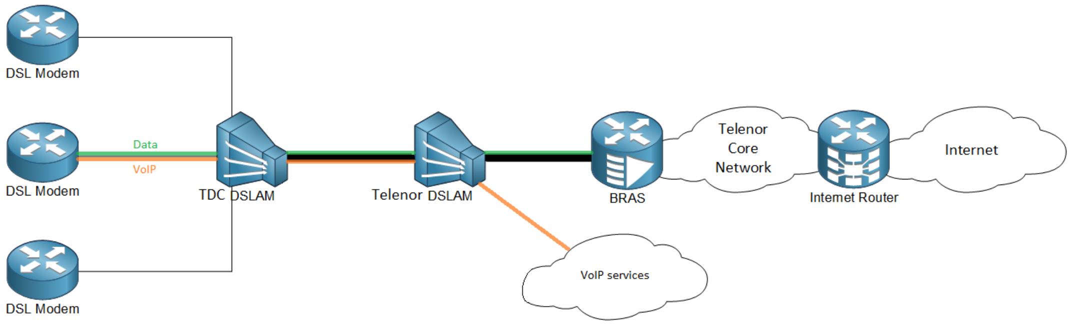


Figure 8.5: A simplified example that presents Telenor's network.

By analysing the network in Figure 8.5, the most obvious[4] bottleneck will be the link between the xDSL modem and the DSLAM. Since users will be connected via Wi-Fi, the wireless medium can be a bottleneck as well. As was shown in Figures 8.6A-D it is very important to be aware of where to apply QoS.

For the upload case, QoS can be implemented in the MAG, before the traffic enters the link between the xDSL modem and the DSLAM. An analysis of the wireless medium is also very important for the upload case, because everyone is competing against each other on equal terms as described for ingress traffic in Section 8.1.2.

[3] Figure 8.5 and Figure A.3 in Appendix A is the same.
[4] It is assumed that Telenor's core network is capable of handling the capacity of their subscribers.

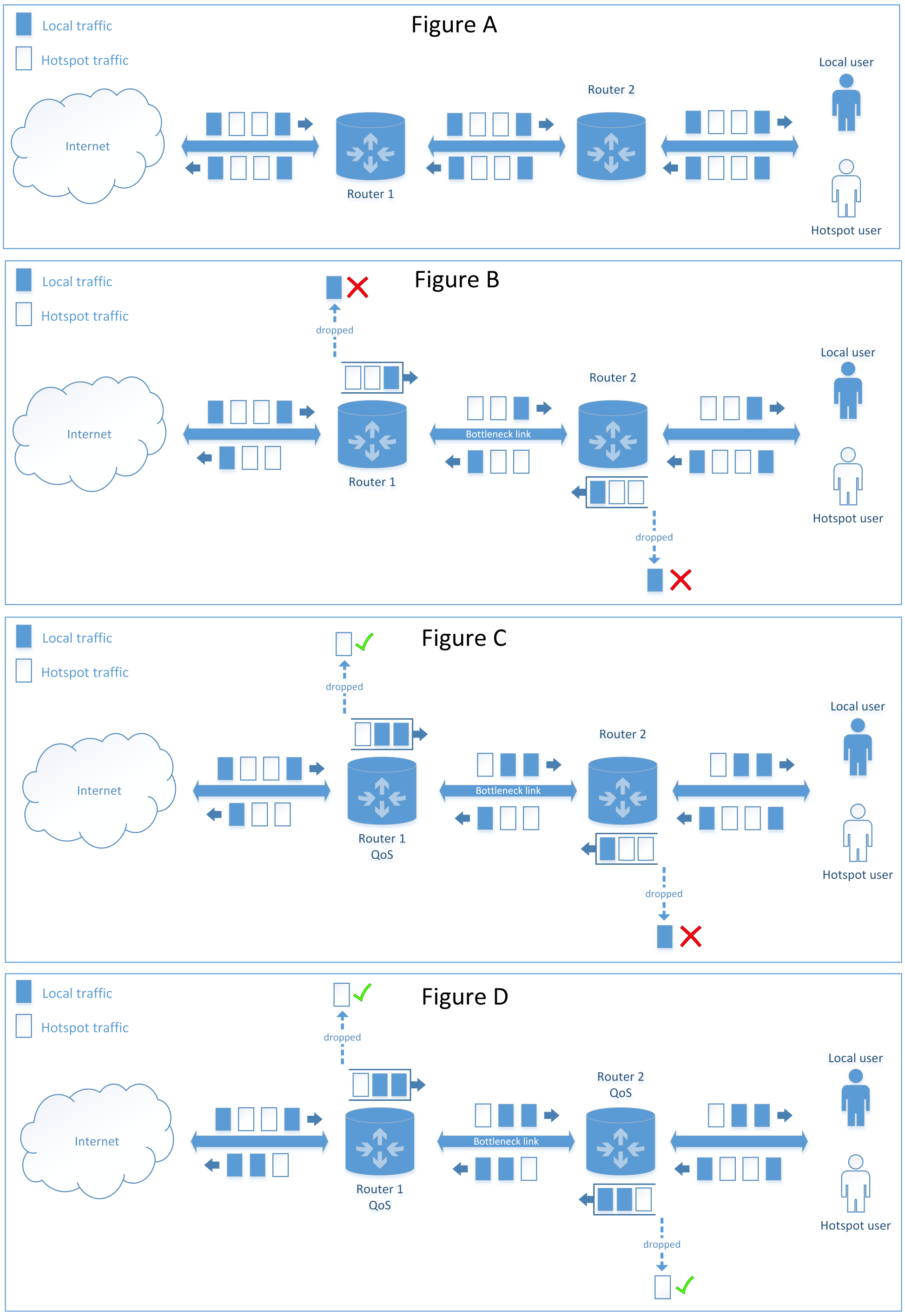


Figure 8.6: Illustration of the *bottleneck link QoS problem* with upload and download traffic

The obvious point to implement download QoS is in the DSLAM, but it can also be done in the BRAS. Telenor has made it clear that it is preferred to implement download QoS in the BRAS, as Telenor would rather have layer 3 QoS over layer 2 QoS. The DSLAM is aware of bandwidth limitations of the link between the xDSL modem and the DSLAM and this information can be forwarded to the BRAS. This can then make the BRAS aware of the bandwidth limitation between the xDSL modem and the DSLAM, which can be applied to the specific Point-to-Point (PPP) session. For the BRAS to apply QoS on download traffic, it is required that it can distinguish between traffic sent by hotspot users and by private users. As all the traffic, at this moment, is behind the same public IP-address, the problem is nontrivial, and an analysis of how to identify download traffic at the BRAS must be done. There are several ways to solve this as for example:

- QoS Tag
  - Ethernet header
  - IPv4 header
- PPP Tunnels
- Port Translation
- Multiple Public IP-addresses

Each of the different solution proposals will be described in the following Sections.

It should be noted that the wireless medium is also an limiting factor for download traffic, especially when also competing with upload traffic. It is however possible to apply download QoS for wireless egress traffic, as explained in Section 8.1.2, but from an theoretical point of view it is not possible to entirely guarantee any level of QoS, as it is based on a probabilistic scheme.

8.2.1 QoS Tag

Adding a VLAN priority in the **Ethernet header** is already used to implement download QoS on the DSLAM for VoIP and data traffic. This priority is used to separate the VoIP traffic to another VPN where the returning traffic will have a higher priority than the regular data traffic received on the DSLAM. Figure 8.5 illustrates the VoIP services, and a similar solution is possible for separating hotspot traffic. This solution does however require new hardware installations which is costly. Telenor is planning to move away from layer 2 aware devices which will make this solution rather short term..

Adding a priority in the **IPv4 header** will require a similar installation, where traffic will be separated on the BRAS, and all hotspot traffic can be forwarded to another VPN. Again this is considered to be a rather costly solution as it requires new hardware.

8.2.2 Point-to-Point (PPP) Tunnel

PPP is a layer 2 point-to-point protocol used to establish a direct connection between two nodes. It can provide authentication, encryption and compression. This is widely used in ISP networks to establish a connection on the xDSL link between a modem and a BRAS. However, there are multiple nodes between the modem and the BRAS which do not support PPP. PPP supports only directly connected point-to-point connections. In order to support PPP connections over multipoint network architecture two protocols are used to create a virtual point-to-point connection. Point-to-Point over Ethernet (PPPoE) and Point-to-Point over ATM (PPPoA) are used to encapsulate the PPP frames respectively to Ethernet frames or ATM frames, depending on the network infrastructure between the two points. As described in Appendix A on page 199 telenor is sometimes part of another ISPs network infrastructure named TDC. TDC is using PPPoE which means that Telenor xDSL subscribers connected to a TDC DSLAM will use a PPPoE connection, while Telenor subscribers connected to a Telenor DLSAM will use PPPoA. TDC DSLAMs are only used if Telenor does not own a DSLAM in the proximity of the subscriber.

Creating another PPP session between each modem and a BRAS to encapsulate hotspot traffic in a separated tunnel to distinguish private traffic, is a reasonable solution to deploy download QoS.

8.2.3 Port Translation

The source or destination port in the header of the transport layer segment can be used to identify traffic, if traffic from each network, hotspot or private, is limited to a known port range. In order to translate several IP addresses into a single public IP address the MAG uses Network Address Translation (NAT) or Network Address and Port Translation (NAPT), with the latter translating both the IP address and the source port. The functionality of NAT and NAPT was explained in Part I Section 3.3.1. The source port after NAPT will remain constant throughout the route to the destined service, and will become the destination port on reply, meaning that upload traffic can be distinguished by investigating the source port and download traffic can be distinguished by its destination port. By translating traffic from each network into specific port ranges it is possible to distinguish which network each packet originates from in order to apply proper QoS at the BRAS.

The MAG has two functions for NAT/NAPT [*AppNote_multipubIPaddress.pdf*, Technicolor [2012b]]: One is *address mapping*, that is a rule of how a single NAT/-NAPT instance is translated. The other function is *address mapping template*, that is a rule, that future NAT/NAPT translations must apply to. Single *address mapping* is not an option as it is impossible to know beforehand which ports will be allocated to whom, which makes *address mapping templates* an interesting option. When a template rule applies an address map is created making *address mapping templates* somewhat dynamic. A template can be added with the command: => *nat tmpladd* which has the following relevant properties:

intf is the interface to which a rule applies to. The interfaces can be found with the command: => *nat iflist* which is seen in Listing 5.16 on page 58.

group is the group to which a rule applies to. A rule must apply to a **intf** or a **group**, but not both.

type is the type of address mapping and can be either *NAT* or *NAPT*.

outside_addr is the IP address after the IP address translation. It could e.g. be the public address of the modem i.e. `85.81.240.13` or `0.0.0.1`, whereas the latter can be used as a wildcard for any address.

inside_addr is the IP address that will be translated. It could e.g. be an IP address belonging to the the hotpsot network.

outside_port is the port or port range that NAPT translates the source port into.

inside_port is the original source port(s) before the NAPT translation.

mode is the mode of the translation. It can be either *outbound*, *inbound*, or *auto*.

When translating ports it is a necessity that the size of the **outside_port** port range is equal to the size of the range specified in **inside_port**[5]. If 412 inside ports in the range *7492-7904* is wished to be translated into a set of outside ports it is necessary to specify a range of 412 outside ports e.g. *40759-41171*. The same rule applies when defining **inside_addr** versus **outside_addr** ranges. An example of a NAPT *address mapping template* is seen in Listing 8.4. The Listing shows how a rule can be added for a client connected to IP address `10.0.0.5`. If the client uses a port in the inside port range it will be translated into a port in the outside port range. The translation is done for the *internet_pppoe* interface as it is the interface going public, which is also why **outside_addr** is `0.0.0.1` i.e. the wildcard for the *first* public IP address, with relation to multiple public IP addresses.

```
=> nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=10.0.0.5 outside_port=40759-41171 inside_port
    =7492-7904 mode=outbound
```

Listing 8.4: Example of how NAPT can be used to translate the outgoing source port, for the public IP address, for the client connected to IP address `10.0.0.5`.

8.2.4 Multiple Public IP-addresses

Providing another public IP-address for the hotspot network in each xDSL modem could be a solution to distinguish traffic, if Telenor can keep track of hotspot assigned

[5] This is a software specific rule, and it is not known why Technicolor has implemented it.

IP addresses versus private assigned addresses. As Telenor has more than 150,000 xDSL subscribers the solution requires Telenor to give out another 150,000 IPv4 IP addresses, one extra for each xDSL subscriber. The solution is possible, but it requires some setup of interfaces in the MAG as can be seen in [*AppNote_multipubIPaddress.pdf*, Technicolor [2012b]]. The configuration is done by defining N public IP addresses with M inside addresses where $M \geq N$. In order for the solution to work the following must be configured:

ATM interface: An ATM interface must be added that connects to the xDSL.

Ethernet interface: An Ethernet interface must be added such that the ATM interface connects to a layer 2 Ethernet interface.

IP interface: The IP interface is created that connects to the Ethernet interface.

Routing: A route must be added, such that there is a route to the Internet.

NAT: A NAT mapping is created such that the M inside addresses are mapped to the N public addresses.

The creation of multiple public IP addresses takes place in the creation of interfaces in Section 3.1 in Part I as well as NAT in Sections 3.3.1 in Part I and 8.2.3.

The extended analysis of the QoS problem and the different possible solutions have identified two independent problems. Upload and download QoS. The following chapter will describe the solution design of each problem based on this analysis.

CHAPTER

NINE

SOLUTION DESIGN

This Chapter will provide a solution design to the upload and download QoS issues presented in Chapter 8, that analysed the internal QoS mechanisms of the MAG and possible bottlenecks in the network observed. This Chapter will provide the requirements in order to ensure that the private user's QoE is not influenced, in any way. The requirements for Part II is as follows:

Quality of Service

Must have

1. Current upload QoS implementations will remain intact.
2. Current download QoS implementations will remain intact.
3. Hotspot users upload traffic will not decrease the xDSL subscribers download or upload resources.
4. Hotspot users download traffic will not decrease the xDSL subscribers download or upload resources.
5. No extra software is required on hotspot user devices and private user devices.

Should have

6. Hotspot users will evenly share the available upload resources.
7. Hotspot users will evenly share the available download resources.
8. The internal resources of the MAG should always favor private users.

As seen the requirements states that hotspot users can only use any spare resources, and that the private user should not, in any way, be able to measure the effect of hotspot users. As the upload and download flow directions are very different from each other the following Chapters, in Part II, will be split into upload and download. The next Sections, describing the solution design for upload and download QoS, will be split as well.

9.1 Upload QoS

This Section will cover the choices made to solve QoS for upload traffic. As mentioned in Section 8.2, the idea is to solve upload QoS in the MAG, since it is assumed that the bottleneck is the link between the xDSL modem and the Digital Subscriber Line Access Multiplexer (DSLAM). The idea is to not interfere with current default QoS implementations for private traffic such as e.g. voice, data and management, and then apply the lowest priority to all hotspot traffic. To describe the design approach it is required to know how hotspot traffic flows through the different elements within the MAG. The description of elements was given in Section 3.1 in Part I, that went through the different elements in order to give a solution to the network integrity problem. The idea is to exploit these elements to assign a low priority to hotspot traffic, such that traffic from private users will *always* be served first. Figure 9.1 shows which elements the traffic flows through for each network, where the hotspot network is marked in yellow and the private network is marked in green.

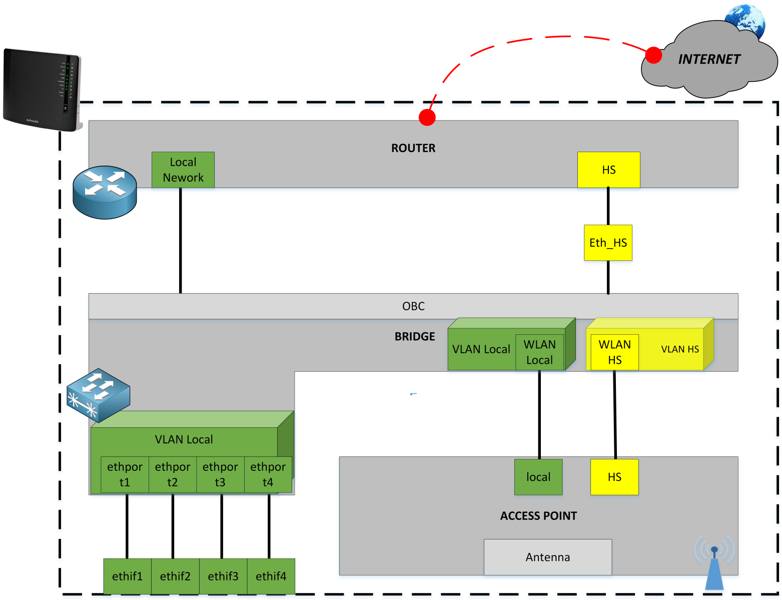


Figure 9.1: The Figure illustrated the flow of traffic for each network. The hotspot network is marked in yellow and the private network is marked in green.

In an effort to avoid unhandled internal congestion e.g. on the OBC bridge port, an internal class should be applied as soon as possible to each traffic stream. The reason for this is because the MAG does resource management, that depends on the internal class assigned[1]. The following list will describe the design choices for the upload QoS starting from the Access Point going all the way to the egress side of the Router:

The Access Point do not directly support any QoS features for upload traffic. The MAG only has one radio, that supports only one wireless channel. It means that both the private network and the hotspot network uses the same wireless channel and will compete for the wireless medium. As described in Section 8.1.2 for the ingress stream, it is possible to assign traffic priorities for traffic on the host station using WMM. It does however require users to install some software that gives all traffic from the host station a certain priority. As the solution should be seamless for the user, no software should be installed, thus no changes for wireless upload QoS will be implemented.

The Bridge has several methods to assign priorities for upload traffic as described in Section 8.1.1. As described the bridge can be configured to use the already assigned priority in the ToS field in the IPv4 header using either one of the **IP precedence field** or **DSCP field** methods. In order to guarantee that the correct ToS priority is given, it requires the user to install extra software on the host station, which is not wished. The most interesting method is the **port default priority** method, that assigns priorities depending on the bridge port the traffic flow belongs to. Another method is the **medium priority** method, which depends on which medium the frame enters from. Because all traffic, hotspot or private, enters from the same medium they will always be assigned the same priority, and this method is thus not an option. As the **port default priority** method is the only applicable method left, this method will be used to assign an internal class to Ethernet frames in the bridge.

The Logical Ethernet Interface can be used to add layer 2 QoS priorities to frames for the Wide Area Network (WAN) side. As it serves no purpose to add a layer 2 QoS priority to the WAN side, for upload traffic (in this project), this will not be implemented.

The Router uses IP QoS to assign layer 3 QoS as explained in Section 8.1.3. The Section described how packets are classified within the router, and the purpose of IP QoS rules and labels. The first thing the router does, when receiving a packet, is to go through the IP QoS rule table until a match is found. As can be seen in Listing 10.2 in Appendix F.2 there are various rules defined for private traffic as well as for e.g. VoIP traffic. In an effort to make sure that hotspot traffic is always classified as hotspot traffic a rule for hotspot traffic will be applied as

[1] As the bridge supports 100 Mbps duplex for each physical interface, it is assumed that elements within the MAG will not be a bottleneck for the 50/10 Mbps download/upload that is the best xDSL subscription that Telenor supports at the moment. However, it will never hurt to apply an internal class for the MAG to do resource management as soon as possible.

early as possible. The rule will use the source interface group *dmz* as a condition to classify traffic from hotspot users, and will be placed in the chain named *qos_user_labels*[2]. The classification method for the label created for hotspot traffic will be set to overwrite the existing internal class priority, even though the traffic has already been assigned with an internal class in the bridge. There is no reason to take a risk, that the internal class is, by fault, not assigned elsewhere, and the internal class it is assigned in the router as well. The internal class priority assigned must go a lower priority queue than private traffic. Adding a QoS parameter in the ToS field in the IPv4 header serves no purpose and will not be assigned for hotspot traffic. The bidirectional option will be enabled in an effort to avoid unhandled internal congestion in the download stream. Inheritance is not needed because the rule created will give any child streams the same label.

The following setup should ensure that upload QoS can be guaranteed for the link between the xDSL modem and the DSLAM. To investigate how upload QoS will affect the wireless medium, and a stress will be done in Chapter 12 after the implementation of IP and Ethernet QoS in Chapter 10.

9.2 Download QoS

This section will cover the design choices made to solve QoS for download traffic. Telenor has made it clear that they wish to do layer 3 QoS over layer 2 QoS, and traffic identification should thereby be done in the BRAS. The problem is, in this project, considered as an identification problem, as an actual implementation must be made on the BRAS that has been inaccessible during the project. It means that the solution should simply provide the BRAS with necessary information, such that it is able to clearly distinction between traffic destined to a hotspot user versus traffic to a private user. As discussed in the previous Chapter, there are several ways to deliver this information. The goal is to create a solution that is robust in terms of Telenor's dynamic network infrastructure, and cheap in terms of resources. The following possible implementations has been investigated:

QoS tag

Using the QoS tags for separating hotspot and private traffic will require new hardware installations which is costly for Telenor, and a solution involving QoS tags will thereby not be implemented.

IP-addresses

Using IPv4-addresses is an expensive solution, as it will require Telenor to distribute another public IPv4 address for each xDSL modem. Even though Telenor,

[2] The chain *qos_user_labels* is always read before the chain *qos_default_labels* that is used for all default rules. The IP QoS configuration guide states that it is best practice to add new IP QoS rules to the *qos_user_labels* chain [*ConfigGuide_IPQoS.pdf*, page 30, Technicolor [2012b]], which makes perfect sense as the classification stops at first match.

at the moment, owns a pool with enough IPv4 addresses, it is clearly not the ideal solution as IPv4 addresses is becoming a limited resource.

PPP tunnels
Using PPP tunnels will require the BRAS to track the double amount of connections, which might require additional hardware that is costly. Another reason for not creating additional PPP tunnels, is because Telenor does in fact pay TDC for each PPP tunnel that goes through a TDC DSLAM as described in Appendix A.

Port translation
The port translation solution does not add additional costs except the work hours it takes for implementation, making it the ideal solution.

The idea is to create port translation rules using NAPT, that assigns hotspot users to a specific port range and private users into a different port range. A port is defined as a 16-bit integer, that goes from 0 to 65535, and some of these ports are reserved for system processes and other have special purpose uses. It means that the port ranges defined must not create conflicts. The entire port range can be split into three standard sub-ranges [IANA [2016]]:

System Ports, also known as the *Well Known Ports*, and goes from 0 to 1023. Port 0 is however not available for communication between hosts.

User Ports, also known as the *Registered Ports*, and goes from 1024 to 49151.

Dynamic Ports, also known as the *Private* or *Ephemeral Ports*, and goes from 49152 to 65535.

To avoid conflicts with system specific ports, it has been chosen to define 5000 to 26847 as the port range for hotspot users. Figure 9.2 illustrates the concept, where traffic from private user can be any of the other ports outside the hotspot user port range.

To provide wireless download QoS the internal queuing system will be used as described for egress traffic in Section 8.1.2. It uses the WMM protocol and the internal class of each frame to provide a probabilistic scheme for wireless download QoS, where frames with high internal class has a higher probability to be sent.

As described in Section 8.1.2 there is a possible issue with upload QoS. All users, hotspot and private, are sharing the wireless medium which in the case of being congested would make any prioritising within the MAG useless (especially for high bandwidth xDSL subscriptions). The prioritising can, as mentioned, be done on the user device (host station), but requires the user to install extra software, which is out of the scope of this project. Since the MAG has only one radio, download traffic might also have an impact on upload traffic, which should be investigated. By doing a stress test of the wireless medium a risk assessment will be made to address how likely the wireless medium it is to become a bottleneck. The stress test will be done in Chapter 12 after the implementation of upload QoS in Chapter 10 and download QoS in Chapter 11.

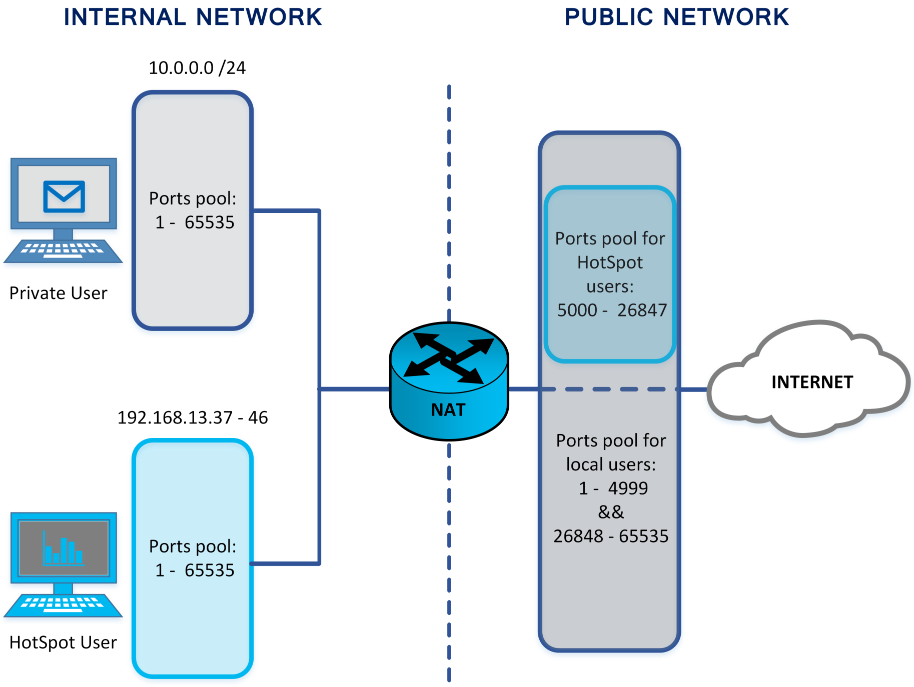


Figure 9.2: The translation of hotspot user ports into the range 5000 to 65535. Private users can use any other port.

CHAPTER

TEN

UPLOAD QOS

This Chapter will implement upload QoS to ensure that traffic from private users is always prioritized over traffic from hotspot users. Section 8.1.2 showed that wireless QoS cannot be ensured without additional software installations on host stations, because all Wi-Fi clients use CSMA/CA when communicating to the access point. The two options left for upload QoS are Ethernet QoS in the bridge instance and IP QoS in the router instance. Ethernet QoS will be implemented as it is the first point (seen from upload), in the MAG, that uses QoS classification in the form of assignment of an internal class. By assigning an internal class as soon as possible ensures that more resources are allocated to private users, as resource management depends on the internal class[1]. IP QoS is implemented in order to make sure that hotspot and private traffic are separated into different queues before the traffic goes to the link between the xDSL modem and the DSLAM. The purpose of IP QoS is to ensure the hotspot traffic is classified into a low priority queue and will only be sent if there is any spare capacity on the xDSL link. Section 10.1 will configure and test IP QoS and Section 10.2 will configure and test Ethernet QoS. At the end of each Section, each configuration will be tested according to the test setup described in Appendix G.

10.1 IP QoS for Upload QoS

An introduction to IP QoS was given in Section 8.1.3 that explained the IP QoS framework in the MAG. This section aims to implement a sustainable IP QoS solution with relation to Section 9.1. Section 10.1.1 will configure IP QoS for upload QoS in the

[1] It is not entirely known if resource management is in fact done outside the router component as no specific documentation has provided any information about this topic. As assignment of internal classes in Ethernet QoS does not introduce any negative effects, it has been chosen to implement Ethernet QoS.

MAG, and Section 10.1.2 will test the performance of the IP QoS implementatio.

10.1.1 Configuration of IP QoS

IP QoS has two functions: It can change the internal class of ingress and egress traffic or it can add a QoS tag to the IPv4 header in the ToS field. Changing the internal class provides local QoS, but can also be used to put traffic into various egress queues. Changing the internal class will result in detectable prioritisation of traffic, especially for upload as mentioned in Section 8.2. The drawback of only changing the internal class is that traffic is not identifiable at Telenor's end of the network, where QoS is also important. The purpose of this Section is to assign proper internal class priorities to traffic from private users such that it will have higher priority than traffic from hotspot users. The **DSCP** parameter, that is part of the ToS field, can be used to add a QoS tag in the ToS field for various traffic, but that option will not be used in this project, as the bottleneck for upload traffic can only be resolved by doing egress queueing in the MAG, and nothing noteworthy can be done at Telenor's side of the network.

Listing 10.1 shows the default IP QoS setup in the MAG gathered with the command => *label list*[2]. The Listing shows the default labels and their assigned internal class as well as their ToS field mark in the IPv4 header (if any).

```
=>label list
Id   Name         Class      Def   Ack [...]  Mark  Type  Value
                                                               [...]
--------------------------------------------------------------------

0    Interactive  increase   8     8          off   tos   0
1    Local        ignore     0     0          off   tos   0
2    Management   increase   12    12         off   tos   0
3    SIPS_RTP     overwrite  14    14         on    dscp  ef
4    SIPS_SIG     overwrite  12    12         on    dscp  af42
5    Video        increase   10    10         off   tos   0
6    VoIP-RTP     overwrite  14    14         on    dscp  ef
7    VoIP-Signal  overwrite  12    12         on    dscp  af42
8    default      increase   default
                                   prioritize off   tos    0
9    voice-only   overwrite  14    14         off   tos    0
```

Listing 10.1: The default IP QoS labels. Columns: *Fapp, Bid, Inh, Use*, and *Trace* has not been shown. The full Listing can be seen in Listing F.1 on page 230.

In order to gain addition insight in the function of each label, Listing 10.2 is shown. The Listing shows the default configuration of the IP QoS ruleset and its relation to the IP QoS labels. Listing 10.2 is a modification of Listing F.2 on page 230, and only shows the most relevant **Chain**s and rule properties. The Listing shows that the chain *qos_user_labels* is currently empty, whereas *qos_defualt_labels* contains various rules

[2] It is not possible to show the whole list directly in the report, but a Listing containing the whole setup can be found in Appendix F in Listing F.1 on page 230.

for different types of traffic. In the *qos_defualt_labels* the most interesting rules, in the **Action** column, are the labels *Interactive* and *default*. By inspecting the rules it is seen that most traffic generated will be classified into the *Interactive* or *default* labels. The **Conditions** for the *Interactive* label are mostly based on ordinary Internet traffic and almost everything that does not match that condition will go to the *default* label.

```
=>label rule list
Rules (flags: C=Constant, D=Dynamic, E=Enable, L=Log)
=====
Chain   Nr.  Action                    Conditions
---------------------------------------------------------
[...]
qos_labels
        [...]
        3    : link qos_user_labels     : *.* > *.*
        4    : link qos_default_labels : *.* > *.*
qos_user_labels
qos_default_labels
        1    : VoIP-Signal             : VoIP-Inc-SIP-UDP wan.* >
                                                               *.*
        2    : VoIP-Signal             : VoIP-Inc-SIP-TCP wan.* >
                                                               *.*
        3    : VoIP-RTP                : VoIP-Inc-RTP wan.* > *.*
        4    : Interactive             : ah *.* > *.*
        5    : Interactive             : esp *.* > *.*
        6    : Interactive             : http *.* > *.*
        7    : Interactive             : httpproxy *.* > *.*
        8    : Interactive             : https *.* > *.*
        9    : Interactive             : imap *.* > *.*
        10   : Interactive             : imap3 *.* > *.*
        11   : Interactive             : imap4-ssl *.* > *.*
        12   : Interactive             : imaps *.* > *.*
        13   : Interactive             : pop2 *.* > *.*
        14   : Interactive             : pop3 *.* > *.*
        15   : Interactive             : pop3s *.* > *.*
        16   : Interactive             : smtp *.* > *.*
        17   : Interactive             : telnet *.* > *.*
        18   : Management              : dns *.* > *.*
        19   : Management              : icmp *.* > *.*
        20   : Management              : icmpv6 *.* > *.*
        21   : Management              : ike *.* > *.*
        22   : Video                   : igmp *.* > *.*
        23   : Video                   : rtsp *.* > *.*
        24   : Local                   : local.* > *.*
        25   : default                 : *.* > *.*
[...]
```

Listing 10.2: The default IP QoS rules before the configuration. The Listing has been modified to only show important options the columns **Flags** and **Rule name** have been deleted. The **Chain**s: *routing_labels*, *rt_user_labels*, *rt_system_server*, *qos_system_service_in*, and *qos_system_service_out* have been removed as well. An unmodified Listing can be seen in Appendix F in Listing F.2 on page 230.

By looking into Listings 10.1 and 10.2 it can be seen that most traffic will be classified to either the *Interactive* (internal class *8*) or the *default* label (internal class is *4*), unless it is some kind of special purpose traffic e.g. VoIP traffic. From Table 8.3 on page 99 it can be seen that internal class value *4* goes to the queue with lowest priority i.e *Best effort*. As it is wished to put hotspot traffic in the lowest priority queue, the *default* label must be given higher priority as much private traffic is assigned to that label. As *Interactive* is already in the *WFQ2* queue, it is possible to put the *default* label into the *WFQ1* queue by assigning it internal class *6* and put hotspot traffic in the *Best effort* queue by assigning it internal class *1*. The following label assignment should be done:

> **HotspotLabel**: Internal class *1* for both **defclass** and **ackclass**. Will go to the *Best effort* queue.
>
> **default**: Internal class *6* for both **defclass** and **ackclass**. Will go to the *WFQ1* queue.
>
> **Interactive** (no changes): Internal class *8* for both **defclass** and **ackclass**. Will go to the *WFQ2* queue.

The values ensures that hotspot traffic goes to the **Best effort** queue, while private user traffic goes to the **WFQ1** queue or higher. VoIP and SIP traffic goes to the *real-time* or *WFQ4* queues and is thereby always prioritised over any other traffic. The configuration can be seen in Listing 10.3.

```
1 label add name=HotspotLabel
2 label modify name=HotspotLabel classification=overwrite defclass=1
      ackclass=1 bidirectional=enabled inheritance=disabled
      tosmarking=disabled fapprio=enabled
3 label rule add chain=qos_user_labels index=1 name=HotspotQoS
      srcintf=dmz log=disabled state=enabled label=HotspotLabel
4 label modify name=default defclass=6 ackclass=6
```

Listing 10.3: Set up QoS with hotspots users having very low priority and xDSL subscribers having higher priority.

Lines 1-2 adds the *HotspotLabel* with **defclass** and **ackclass** *1*. **bidirectional** is *enabled* as the same internal class will be given to download traffic when it returns. **inheritance** is *disabled* as it does not change anything. *Inheritance is used to give the same internal class to child connections, but all connections will, in this case, always originate from the same interface i.e. dmz*. **tosmarking** is *disabled* as no ToS will be added to the IPv4 header. **fapprio** is *enabled* such that resources will be shared fairly among users.

Line 3 adds the label rule in the *qos_user_labels* **chain**. It is added to that chain as it will be checked before the *qos_default_labels* **chain**. The **Conditions** is added for the *dmz* **srcintf** as all hotspot users belongs to the **group** *dmz*[3].

[3] Two IP interfaces were made in Chapter 5 on page 49: The **LocalNetwork** IP interface for the private network, belonging to group **lan**, and the **HotSpotIP** IP interface for the hotspot network, belonging to group **dmz**.

Line 4 changes the internal class of the *default* label in order to move it to the *WFQ1* queue.

The IP QoS labels created can be seen in Listing 10.4 and the new rule using the *HotspotLabel* label is seen in Listing 10.5. Both Listings has been modified and an unmodified version can be seen in Appendix F in Listings F.3 on page 233 and F.4 on page 234.

```
=> label list
Id  Name        [...]  Def  Ack   Fapp Bid Inh Mark Type Value [...]
--  ----               ---  ---   ---- --- --- ---- ---- -----
10  HotspotLabel 1     1     on   on  off off  tos  0
0   Interactive  8     8     off  off off off  tos  0
11  LanPQ        14    14    off  off off on   dscp ef
1   Local        0     0     off  off off off  tos  0
2   Management   12    12    on   off off off  tos  0
3   SIPS_RTP     14    14    off  on  off on   dscp ef
4   SIPS_SIG     12    12    off  on  off on   dscp af42
5   Video        10    10    off  on  off off  tos  0
6   VoIP-RTP     14    14    on   on  off on   dscp ef
7   VoIP-Signal  12    12    on   on  off on   dscp af42
8   default      6     6     off  off off off  tos  0
9   voice-only   14    14    off  off on  off  tos  0
```

Listing 10.4: The newly created *HotspotLabel* and the modified *default* labels can be seen in the Listing. The coloumns *Class*, *Use* and *Trace* has been deleted to make the list fit, but can be found in Listing F.3.

```
=> label rule list
Rules (flags: C=Constant, D=Dynamic, E=Enable, L=Log)
=====
Chain    Nr. Rule Name   Action     Conditions
-----    --- ---------   ------     ----------
[...]
qos_user_labels
         1   HotspotQoS  HotspotPQ  dmz.* > *.*
[...]
```

Listing 10.5: IP QoS rules for hotspot belonging to the **group** *dmz*. The unmodified Listing can be seen in Listing F.4

10.1.2 Test of IP QoS

In order to test the IP QoS configuration a test is performed, that takes basis in the setup described in Appendix G on page 237. The test is performed with one hotspot user and one private user competing for upload capacity. The ideal scenario is that the hotspot user will not gain any capacity if the private user takes it all, as hotspot users can only use any spare capacity. During the test both users are doing a throughput test using UDP to a server located at AAU and the start time of the private user is slightly delayed. It is assumed that the bottleneck lies at the xDSL link between the xDSL modem and the DSLAM and that the AAU network has higher capacity than that link. The test scenario is illustrated in Figure 10.1.

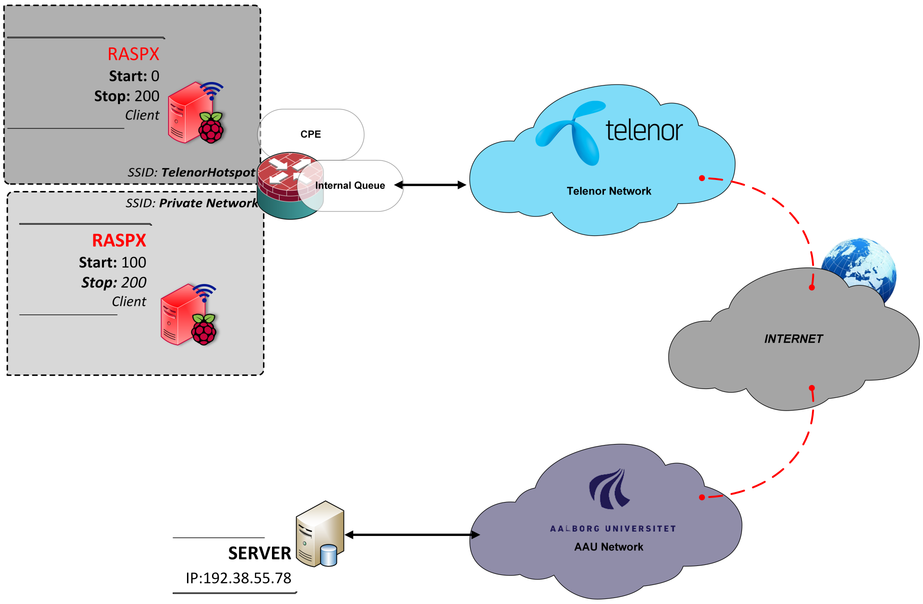


Figure 10.1: A hotspot user and a private user are doing a throughput test for upload at the same time. The test is performed to a server located at AAU and the start time of the private user is slightly delayed.

Figure 10.2 illustrates a throughput test done with two Rasberry Pi devicess. One is connected to the hotspot network and starts first. It is seen that it quickly reaches a bandwidth of around $1.8\,\mathrm{Mbps}$, which is the maximum upload capacity of the xDSL link. After some time a private user initiates a throughput test and it is seen that it gains almost all of the capacity at around $1.8\,\mathrm{Mbps}$ after some time has passed. The result from the test shows that it is possible to let the hotspot user use only the spare capacity by changing internal classes for different interfaces.

As IP QoS does not guarantee QoS throughout the MAG, but only for layer 3 traffic, Ethernet QoS should be implemented as well. The idea behind Ethernet QoS is to assign an internal class already at the bridge (layer 2) such that internal bottleneck inside the MAG might be avoided and eventual resource management is assigned to traffic as soon as possible.

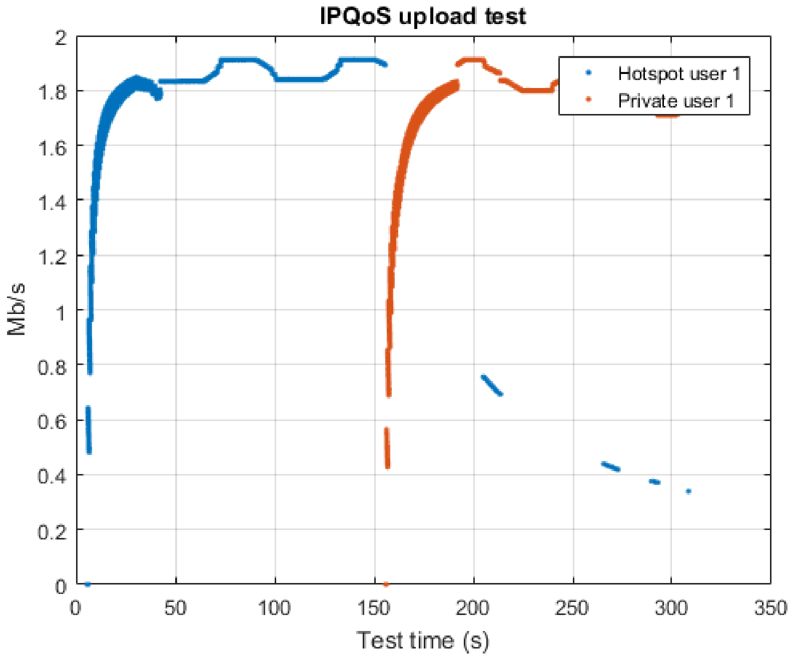


Figure 10.2: UDP wireless QoS upload bandwidth test by two Rasperri Pi devices

10.2 Ethernet QoS

As noticed from the upload QoS performance test from IP QoS, IP QoS is perfectly capable of providing a clear prioritisation of traffic from private users that meets the requirements. One might thus wonder why Ethernet QoS is implemented. There are, however, other factors which can have an effect on the bandwidth such as the internal memory resources in the router component. Queuing packets in the right queue is fine, but if the router is not fast enough to do this, because of internal congestion and resource starvation, it will have a negative effect on the bandwidth. Thus Ethernet QoS will be used in an attempt to reserve internal resources for private users in case of resource starvation.

Section 10.2.1 will configure Ethernet QoS in the MAG, and Section 10.2.2 will test the setup.

10.2.1 Configuration of Ethernet QoS

Before starting to change any of the configurations it is necessary to know what the default settings looks like. Listing 10.6 shows the default bridge port settings for **WLAN**, which is the port used for private wireless traffic, and the physical Ethernet bridge ports **ethportX**, where X can take the values *1-4* as there are 4 physical Ethernet ports. Finally the **HotspotBridge** bridge port is shown, which is the the port used for the wireless hotspot network. The relevant settings for Ethernet QoS are marked in orange and the properties of the settings **Priority Config**, **IP Prec**, and **Priority** was explained in Section 8.1.1.

```
=>eth bridge iflist
[...]
ethportX  : dest              : ethifX
            Connection State: connected   Retry: 10
            Priority Tagging: Disabled
            PortState       : forwarding  Interface: down
            PortNr          : X
            multiWANuntagged  : disabled
            Multicast filter: disabled
            WAN             : disabled
            IGMP snooping   : enabled
            MLD snooping    : enabled
            Transparent Prio: disabled
            BPDU Filtering  : disabled
            Extra Tagging   : none
            Dynamic VLAN    : disabled
            VLAN: Default VLAN: default  Ingressfiltering: disabled
                  Acceptvlanonly: disabled
                  Priority Config:  disabled IP Prec:  disabled Priority:  0
                  Regeneration table    : 0 1 2 3 4 5 6 7
            RX bytes: 0          frames: 0
            TX bytes: 0          frames: 0            dropframes: 0

WLAN:
        dest            : wlif1
        Connection State: connected   Retry: 10
        Priority Tagging: Disabled
        PortState       : forwarding  Interface: up
        PortNr          : 5
        multiWANuntagged: disabled
        Multicast filter: disabled
        WAN             : disabled
        IGMP snooping   : enabled
        MLD snooping    : enabled
        Transparent Prio: disabled
        BPDU Filtering  : disabled
        Extra Tagging   : none
        Dynamic VLAN    : disabled
        VLAN: Default VLAN: default  Ingressfiltering: disabled
             Acceptvlanonly: disabled
             Priority Config:  disabled IP Prec:  disabled Priority:  0
             Regeneration table    : 0 1 2 3 4 5 6 7
        RX bytes: 2241195    frames: 20869
        TX bytes: 34385417   frames: 30076        dropframes: 0

HotspotBridge:
            dest            : wl_ssid1_local0
            Connection State: connected   Retry: 10
            Priority Tagging: Disabled
            PortState       : forwarding  Interface: up
            PortNr          : 9
            multiWANuntagged: disabled
            Multicast filter: disabled
            WAN             : disabled
            IGMP snooping   : enabled
            MLD snooping    : enabled
            Transparent Prio: disabled
            BPDU Filtering  : disabled
            Extra Tagging   : none
            Dynamic VLAN    : disabled
            VLAN: Default VLAN: HotspotVLAN  Ingressfiltering:
                disabled
                  Acceptvlanonly: disabled
                  Priority Config:  disabled IP Prec:  disabled Priority:  0
                  Regeneration table :  0 1 2 3 4 5 6 7
            RX bytes: 0          frames: 0
            TX bytes: 0          frames: 0            dropframes: 0
[...]
```

Listing 10.6: The wireless bridge port settings for the private network.

By looking at the settings, the wired Ethernet bridge ports **ethportX**, the wireless bridge port for the private network **WLAN**, and the hotspot bridge port **HotspotBridge**, it is seen that they are configured in the same way. They all use the medium priority, which gives all traffic the internal class value *4*. To make sure that private traffic will always have a higher priority than hotspot traffic, the **HotspotBridge** bridge port will be configured to use the evaluated priority and to use the 802.1p user priority by assigning *overwrite* to the **Priority Config** setting. By giving the value *disabled* to **IP Prec**, it uses the default port priority i.e. the value in **Priority**. All traffic arriving to the **HotspotBridge** bridge port, with no VLAN tag, will with that configuration be assigned the default port priority as the internal class. In the upload case there is no VLAN tag on frames arriving at a bridge port since this is added by the Logical Ethernet Interfaces. By assigning the default port priority value *1* (internal class value *0* according to Table 8.1 on page 91) to the **HotspotBridge** bridge port, private traffic will always have a higher internal class as it uses the medium priority that assigns internal class value *4*. The configuration of *HotspotBridge* can be seen in Listing 10.7.

```
eth bridge ifconfig intf=HotspotBridge prioconfig=overwrite
eth bridge ifconfig intf=HotspotBridge priority=1
```

Listing 10.7: Commands to apply changes

The reason the default priority is set to *1* is because this maps to the lowest internal class as seen in Table 8.1. The changes made can be seen in Listing 10.8, that shows the settings for the **HotspotBridge** bridge port.

```
=>eth bridge iflist intf=HotspotBridge
HotspotBridge:
            dest            : wl_ssid1_local0
            Connection State: connected   Retry: 10
            Priority Tagging: Disabled
            PortState       : forwarding  Interface: up
            PortNr          : 9
```

```
multiWANuntagged: disabled
Multicast filter: disabled
WAN             : disabled
IGMP snooping   : enabled
MLD snooping    : enabled
Transparent Prio: disabled
BPDU Filtering  : disabled
Extra Tagging   : none
Dynamic VLAN    : disabled
VLAN: Default VLAN: HotspotVLAN  Ingressfiltering:
    disabled
      Acceptvlanonly: disabled
      Priority Config:  overwrite IP Prec:  disabled Priority:  1
      Regeneration table :  0 1 2 3 4 5 6 7
RX bytes: 0          frames: 0
TX bytes: 0          frames: 0            dropframes: 0
```

Listing 10.8: Hotspot bridge port settings

10.2.2 Test of Ethernet QoS

To see the impact of these changes, a throughput performance test is made according to Appendix G. In order to see the impact of Ethernet QoS, the IP QoS settings made in Section 10.1 are temporarily set to use the internal class assigned in Ethernet QoS. The changes made in IP QoS before doing this test is seen in Listing 10.9.

```
label modify name=HotspotLabel classification=ignore
label modify name=Interactive classification=ignore
label modify name=default classification=ignore
```

Listing 10.9: Commands to remove IP QoS for traffic from the hotspot network.

Figure 10.3 shows the result of the throughput performance test, when Ethernet QoS is assigned. It is seen that the hotspot user loses almost all bandwidth as soon as the private user starts transmitting. The conclusion is that upload QoS can in fact be provided by using either Ethernet QoS or IP QoS or both.

It can be concluded that for upload QoS there is no difference whether the prioritisation is done on layer 2 or layer 3, the results are the same. But for resource management purposes it is desired to provide it as early as possible, and the same priority provided in layer 2 will be provided in layer 3 as well.

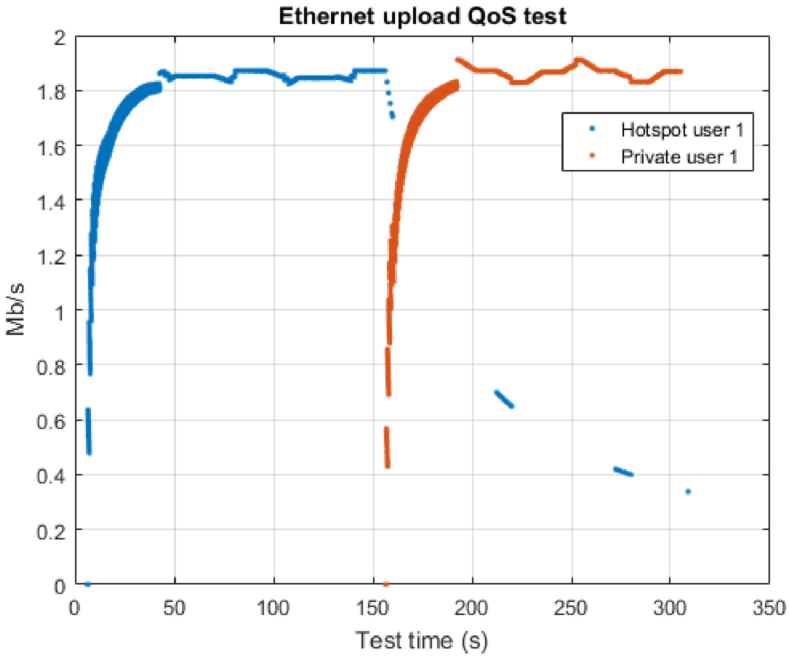


Figure 10.3: A hotspot user and a private user is doing a throughput upload test for 300 seconds. The private user starts transmitting after 150 seconds.

CHAPTER

ELEVEN

DOWNLOAD QOS

This Chapter will implement download QoS to ensure that the xDSL subscriber is always prioritized over hotspot users in the downstream flow direction. Chapter 10 implemented Ethernet QoS and IP QoS in order to ensure that the private user(s) is always prioritised first for upload traffic. The IP QoS label for hotspot users were given the property *bidirectional=enabled*. This property makes it such that when traffic goes back (download), hotpsot users it will be assigned the same label. It ensures that no IP Qos or Ethernet QoS need to be made for hotspot download traffic in the MAG, because an internal class with value *1* is indirectly assigned.

The next step it to add QoS before the xDSL link bottleneck as described in Section 8.2. The Section went through different solutions and it was discovered that download QoS can be assigned at either the DSLAM or the BRAS. As Telenor prefers layer 3 QoS the BRAS it the best option for download QoS. Section 9.2 mentioned that the best option to identify traffic at the BRAS is to translate hotspot traffic into a specific port range. Section 11.1 will explain how port translation is configured in the MAG in order to classify traffic at the BRAS. Section 11.2 will implement wireless QoS for egress traffic.

11.1 Port Translation for Download QoS

Section 9.2 described that Telenor is moving towards layer 3 QoS over layer 2 QoS, which makes port inspection at the BRAS the ideal choice. The BRAS can do port inspection for both upload and download traffic, but it will only be implemented for download traffic as the bottleneck has already been passed, at the BRAS, for upload traffic. In order for the solution to work for download traffic the destination port will be inspected at the BRAS. To limit the project scope it has been chosen to focus on configuration of the MAG, and *port inspection will not be implemented at the BRAS*. It is known that the BRAS model used by Telenor is able to do port inspections, making the solution very useful. The purpose of this Section is to show how NAPT in the MAG can be used to do port translations as well as test the solution.

A specific port range will be chosen for hotspot users, and everything inside that specific port range will never be prioritised over other traffic. It is desirable for the port range chosen to be outside the system ports port range i.e `0-1023`. By observation is has been discovered that the NAPT in the MAG translates ports by default into the range `49152-65535` and it is thereby also desirable to be outside that range. As described in Seciton 9.2, it is decided that hotspot users will be moved to the port range `5000-26847` as it is outside both port ranges just mentioned. In Part I Chapter 5 it is defined that 8 hotspot users can be connected simultaneously, and are can use the following range of IP addresses: `192.168.13.[37-44]`.

11.1.1 Configuration of Port Translation

As it is desired to translate 65535 ports into an range of $26847 - 5000 = 21847$ ports, some extra effort is needed, as translation of ports has to be done one-to-one as explained in Section 8.2.3. A range of **inside_port**s needs to be mapped to an equal range of **outside_port**s. Figure 11.1 shows how it is decided to translate all possible hotspot ports into the range `5000-26847`. It is done by mapping everything from the range *1-4999* into *5000-9998*, *26848-46191* into *5000-24343*, and *46192-65535* into *7504-26847*. A fourth NAPT template, *5000-26847* into *5000-26847*, ensures that **inside_port**s will not be translated into the default port range used by the private network i.e. the port range `49152-65535`.

Just as with port translation, a size *N* **inside_addr**es needs to be mapped into an equal size *N* **outside_addr**es. It raises a minor problem as there is only one wildcard for the public address i.e. *0.0.0.1*, but no wildcard for all hotspot addresses. It means that the four NAPT templates must be created for each internal hotspot IP address. With 8 addresses it results in a total of 32 new templates.

By issuing the CLI command => *nat iflist* the IP interfaces can be seen as shown in Listing 5.16 on page 58. When traffic is routed external it will always go through either the *internet_pppoe* or the *internet_pppoa* interfaces[1]. Which interface is being used depends on which PPP method is employed, this project uses only the *internet_pppoe* interface, because the xDSL connection used in this project is connected to a TDC DSLAM (see Appendix A on page 199). The NAPT templates will be added to the *internet_pppoe* interface as seen in Listing 11.1. The Listing uses X as a variable for which IP address the rule is made for. X can take values between *37* and *44* to make a template for each of the IP addresses in the range `192.168.13.[37-44]`.

[1] TDC uses PPPoE and direct Telenor uses PPPoA or PPPoE.

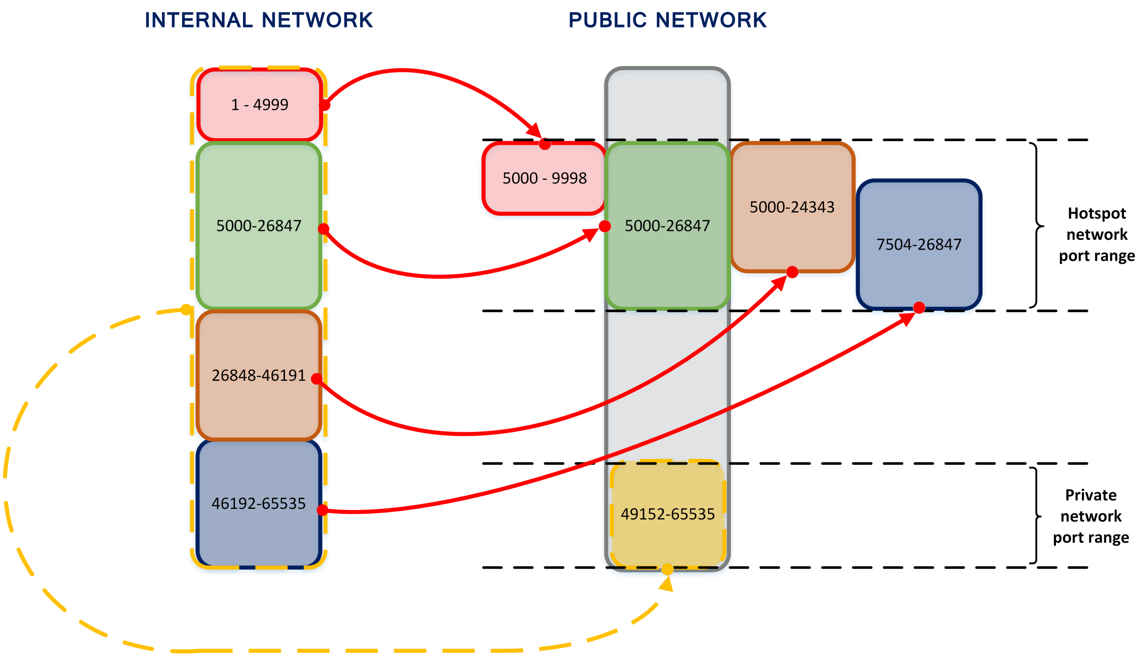


Figure 11.1: The Figure shows how a specific port range is chosen and all ports are mapped into that area. Following the rule of one-to-one mapping.

```
nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=192.168.13.X outside_port=5000-9998 inside_port
    =1-4999 mode=outbound status=up description=HSXRule1
nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=192.168.13.X outside_port=5000-24343 inside_port
    =26848-46191 mode=outbound status=up description=HSXRule2
nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=192.168.13.X outside_port=7504-26847 inside_port
    =46192-65535 mode=outbound status=up description=HSXRule3
nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=192.168.13.X outside_port=5000-26847 inside_port
    =5000-26847 mode=outbound status=up description=HSXRule4
```

Listing 11.1: The four NAPT templates that translates hotspot users' ports into a specific range. X can take values between 37 and 44.

As mentioned, no templates will be added for private users, as NAPT by default translates everything into the *dynamic ports* range i.e. `49152-65535`. As port translation templates have now been added to the *internet_pppoe* interface, as traffic is going external, a test will be performed in the next section to ensure that the NAPT will always perform as expected.

11.1.2 Test of Port Translation

To check whether NAPT does what is expected from the configuration in Listing 11.1 a test will be performed. The test setup is illustrated in Figure 11.2 and shows various hotpsot and private users that connect to a server. The idea behind the setup is to store the original source port and compare it to the source port received at the server after it has been translated by NAPT. The original port are chosen at random in the range `1-65535` and some is chosen by the OS. The test is performed for hotspot users as well as private users to ensure that no wrong classifications can happen.

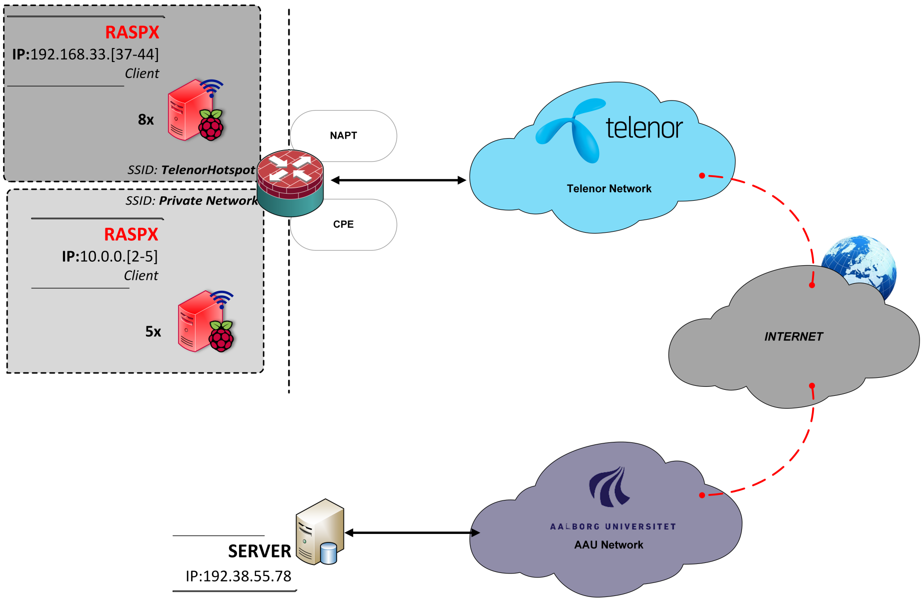


Figure 11.2: The test setup is illustrated with both hotspot and private users connected. The server is located at AAU and has a public IP address.

Figure 11.3 shows the result of the test. The test compares the received port at the server with the original port. The test shows that private users is located in the *dynamic ports* range and hotspot users are always mapped to the port range `5000-26847`.

Figure 11.3 approves that the NAPT setup works and it should be able to implement download QoS for the system at the BRAS. As the scope of the project has been limited, no configuration and test of Telenor's BRAS will be made. It has been told that the BRAS can do QoS management based on port inspection, and the solution should thereby work.

As the xDSL link is not the only download bottleneck in the system, for high capacity xDSL links, the next Section will implement wireless QoS for download traffic.

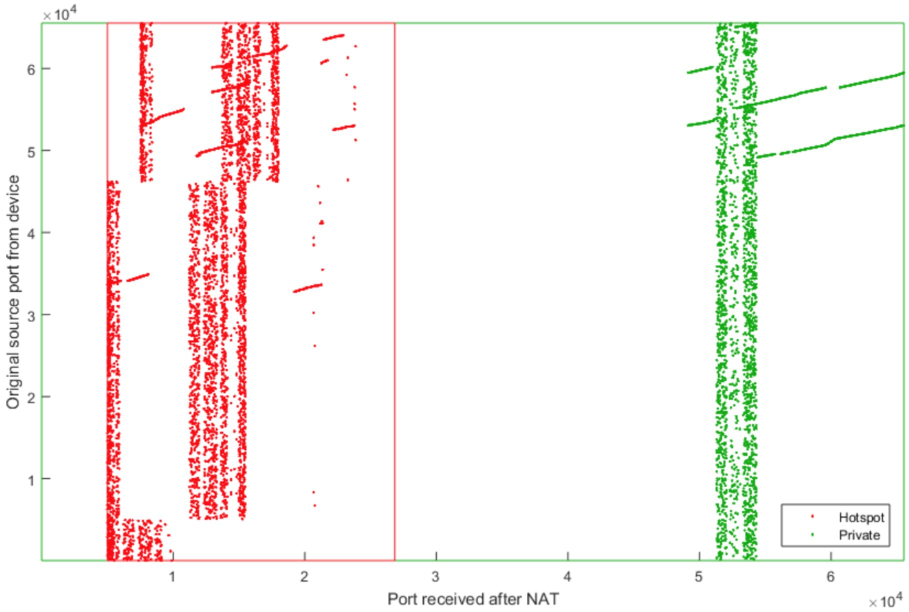


Figure 11.3: The result of the test. Hotspot ports are shown in red and should be inside the red area and private ports are shown in green and should be outside the red area.

11.2 Wireless QoS

An introduction to wireless QoS for egress traffic was given in Seciton 8.1.2. Wireless QoS is by default enabled, in the MAG, and no other configuration is needed as the ACS is chosen based on internal class that has been configured in Chapter 10, given that *bidirectional=enabled*.

Download and upload QoS has now been enabled by all means in the MAG. The configuration should ensure that the xDSL link bottleneck has been prioritised to always serve private users first. Wireless download QoS is also enabled, and ensures that the wireless connected private users are served first for download traffic. Wireless upload QoS does however introduce a problem as all wireless connected devices competes for the wireless medium on equal terms. Chapter 12 will analyse the effect of wireless devices connected to both the hotspot network and to the private network. The aim of the chapter is to analyse how wireless connected devices competes for the medium in the upload and download cases, and how each case affects the other.

CHAPTER

TWELVE

WIRELESS QOS

This Chapter will go through an extended stress test of the wireless medium and the radio of the MAG in order to investigate how the QoS implementation performs in wireless bottleneck scenarios. This is to see how likely it is that the wireless medium will be a limitation towards ensuring QoS for the private user. The Chapter will follow the following structure:

- Upload Tests in Section 12.1.
- Download Tests in Section 12.2.
- Simultaneous upload and download Tests in Section 12.3.
- Result Summary in Section 12.4.

The first Section will be testing upload scenarios followed by a Section for download scenarios. Then a Section will test the wireless medium with both download and upload traffic to see how they affect each other, followed by a summary of the test results. In order to grasp the idea of what the test will look for a basic test scenarios, for the sections, are shown in Figure 12.1.

In some of the Figures that presents the results, e.g. Figure 12.2, periodical spikes can be observed. It is not entirely known what causes these spikes, but it is known that they happen due to unexplained behaviour of the Raspberry Pi devices, and not because of the network, the test setup, or the configuration of the MAG. This behavior should therefore be neglected. All the tests will use the test program described in Appendix G on page 237, and it should be emphasized that all tests are performed using the UDP protocol in order to avoid TCP congestion control and what follows. As the tests are performed in a group room at AAU, it can not be guaranteed that the channel used is

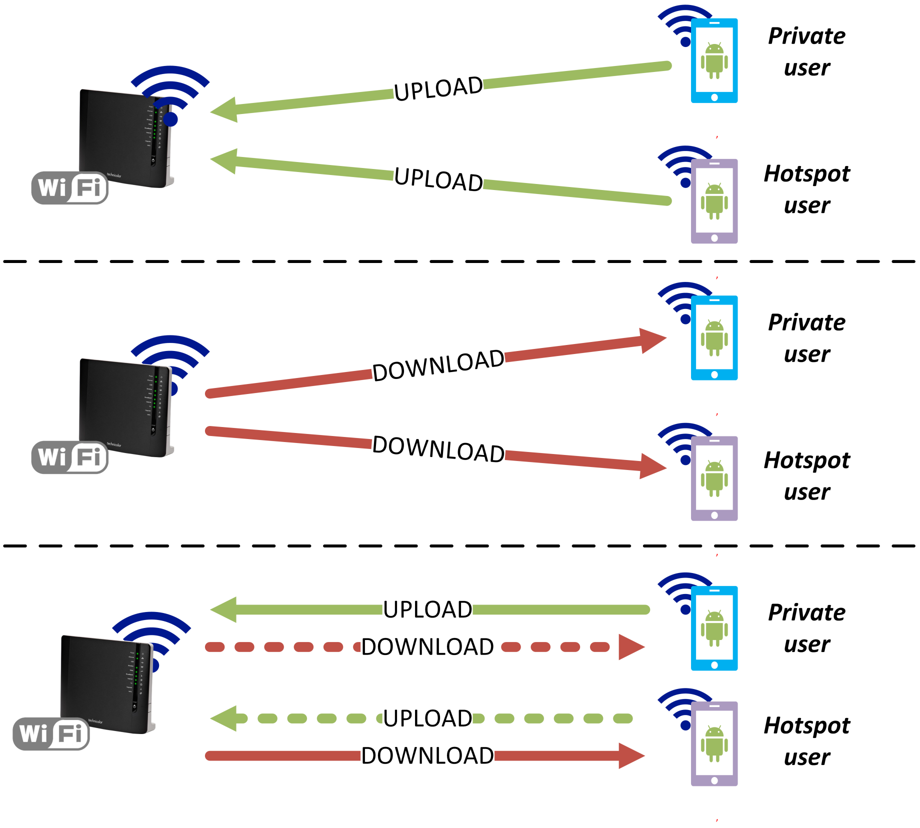


Figure 12.1: Basic test scenarios that are testing separately wireless upload and download with private and hotspot users as well as relationship between simultaneous download and upload test.

not interfered by other devices and/or access points, which might influence the results, meaning that no tests are performed in a controlled environment.

12.1 Upload Tests

The first test provides insight into how several users compete against each other when all attempts to upload as much as possible simultaneously. The test is made with three Raspberry Pi devices connected to the private network. The test is performed over 300 seconds, where each device is delayed 100 seconds from each other. Figure 12.2 shows the result of the experiment. As it can be seen, with each new user, the total throughput is evenly shared between all the users. This behaviour was excepted as all users have equal rights to the wireless medium (CSMA/CA) and thereby shares it. From the test it can be concluded that the capacity of the wireless link is around 40 Mbps, which is shared across all wireless connected users. Similar behavior has been observed with only hotspot users performing upload tests on the wireless link.

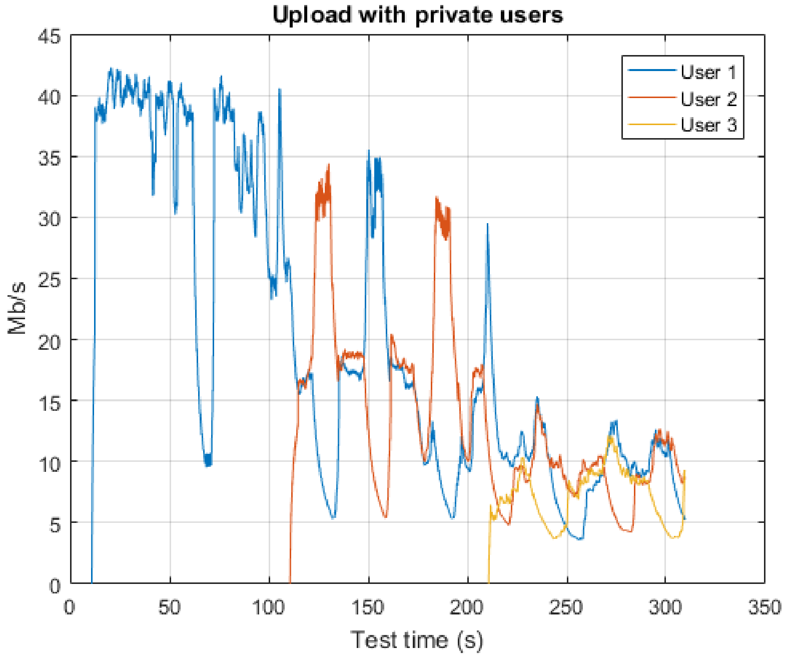


Figure 12.2: Upload test with three private users delayed from each other

The next test, shown in Figure 12.3, shows users from both the hotspot and the private network, that performs throughput tests, for upload, at the same time. The test is made in order to examine how hotspot users compete against private users during upload on the wireless medium. The test is performed over 300 seconds, where a private user start transmitting alone, followed by hotspot users that starts transmitting respectively after 100 and 200 seconds delay.

It is noticed that all users will evenly share the available bandwidth on the wireless medium during upload, independent from which network they belong to. As no upload QoS can be set up on the client devices, traffic from different SSIDs can not be prioritized, and all users are thus sharing the bandwidth evenly. This can, in the case of a large number of users, cause a problem since private users can possibly experience degradation of guaranteed throughput as it can not be assured by the QoS.

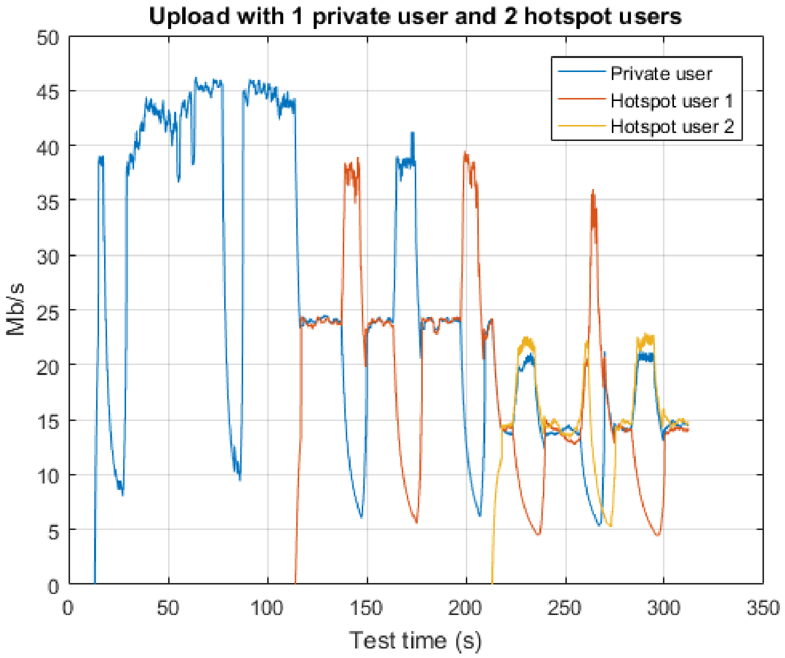


Figure 12.3: Upload test with one private user and two differently delayed hotspot users.

12.2 Download Tests

As for upload, three private users will now compete for the download traffic on the wireless medium. Figure 12.4 shows the results. As noticed the capacity is evenly shared.

The next test is performed with two hotspot users and one private user. *Hotspot user 1* starts downloading followed by *Hotspot user 2* 100 seconds later, and then a private user starts 100 seconds later than that. Figure 12.5 shows the result of the test. The result shows that hotspot users can not utilise the full capacity, because their traffic is in a queue with very low priority, and thus the radio may wait a longer time before trying to transmit. When the private user starts downloading, the traffic from the hotspot users are decreased rapidly. It can be seen that download QoS can be guaranteed on the wireless medium.

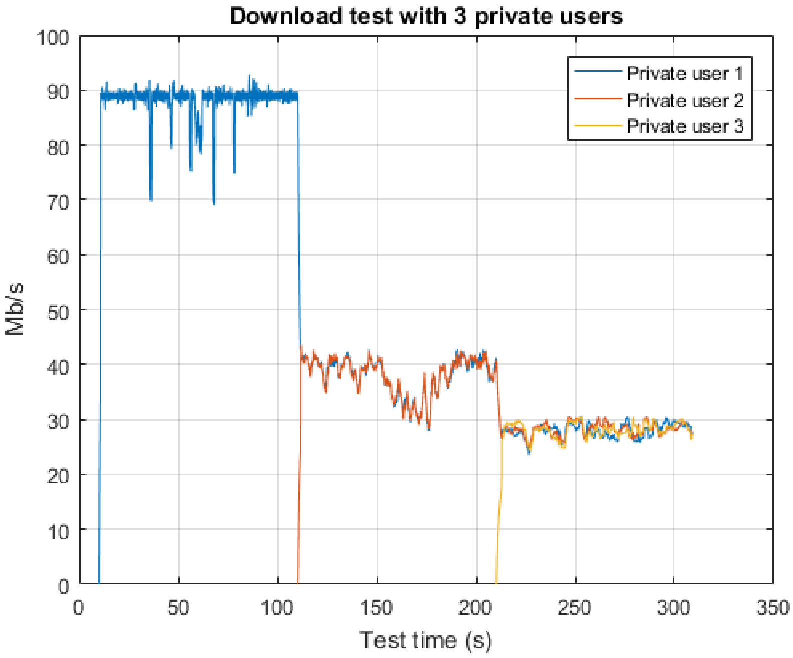


Figure 12.4: Download test with three private users delayed from each other.

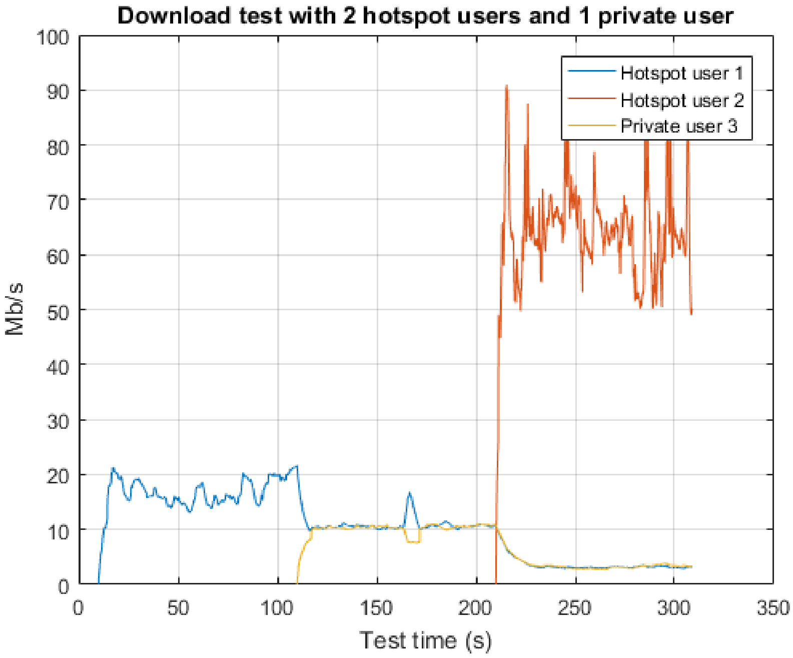


Figure 12.5: Download test with two hotspot users and a private user delayed from each other.

12.3 Simultaneous Upload and Download Tests

This Section will be testing close to real world scenarios, where the wireless medium is used for both download and upload traffic at the same time. The tests are made to learn how well the radio ensures download and upload QoS for the private user in cases where there is a lot of the opposite direction traffic from the hotspot network. The reason these tests are interesting is because WiFi is half duplex, which means that upload and download is sharing the overall capacity. Figure 12.6 illustrates a test with one uploading private and hotspot user, followed by a downloading private user 100 seconds later. As seen the upload speed of both the hotspot user and the private user is heavily affected by the downloading private user. By comparing the download throughput with Figure 12.4 it shows that the same throughput is not achieved. The test shows that uploading hotspot users can have a negative impact on the download bandwidth of the private user. QoS has been applied for download traffic and the radio will give download traffic a higher probability of transmission on the shared wireless medium, compared to the transmission probability the devices gives the upload traffic.

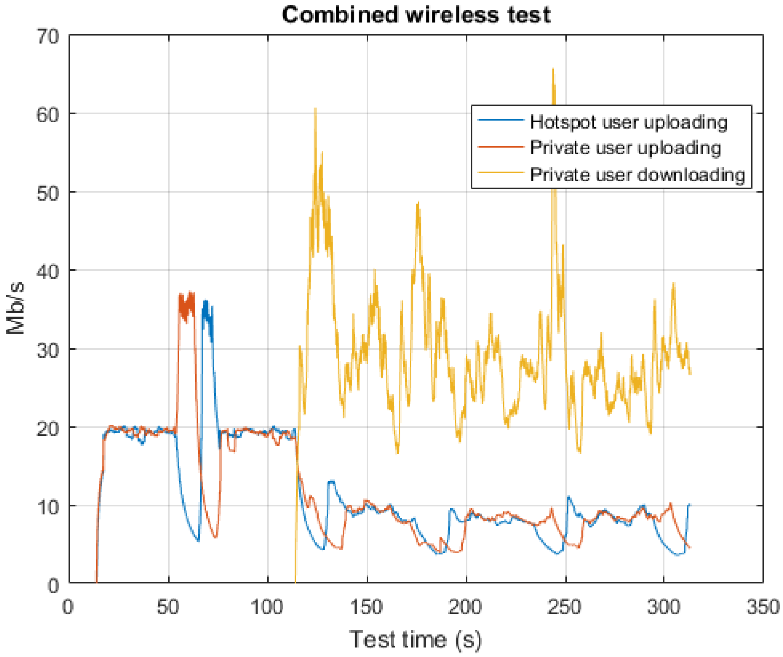


Figure 12.6: A hotspot user and a private user starts uploading, followed by a private user, who starts downloading with a delay of 100 seconds.

Figure 12.7 shows the test results of a hotspot user and a private user uploading, followed by a hotspot user that downloads. The test is made is to measure how a downloading hotspot user will affect the upload bandwidth of private users. The Figure shows that the low priority download traffic has a very low chance of transmission compared to the upload traffic. This means that downloading hotspot users will have relatively little impact on private or hotspot users upload bandwidth.

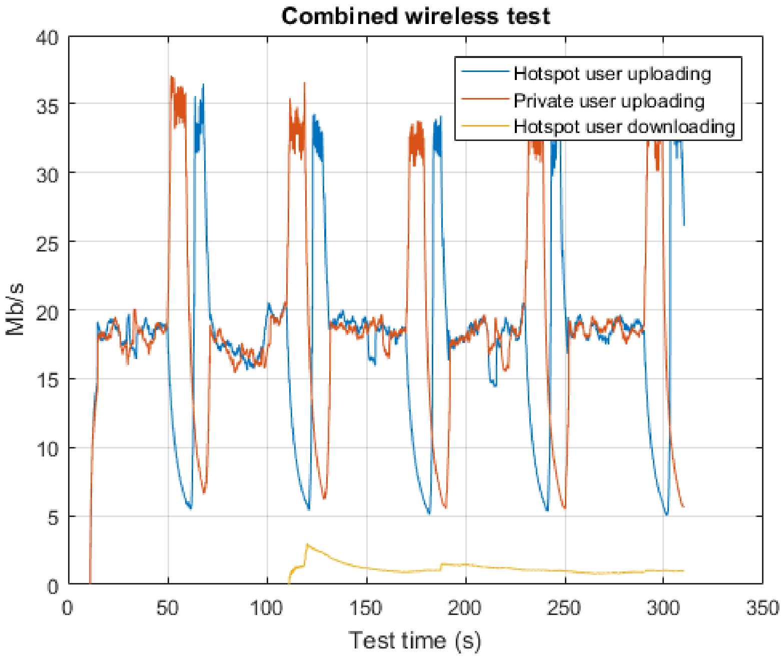


Figure 12.7: A hotspot user downloads and two private users that are uploading. The downloading hotspot user is delayed by 100 seconds.

A third test is made to further investigate if downloading hotspot users has an impact on a private users upload bandwidth. The test is made with one private user uploading, followed by three downloading hotspot users that are delayed by 100 seconds. The result of the test can be seen in Figure 12.8 on the next page and shows that the private users upload bandwidth is slightly changed when the three hotspot users starts downloading. This impact is however so small that changes on these scales can be considered undetectable on a practical level.

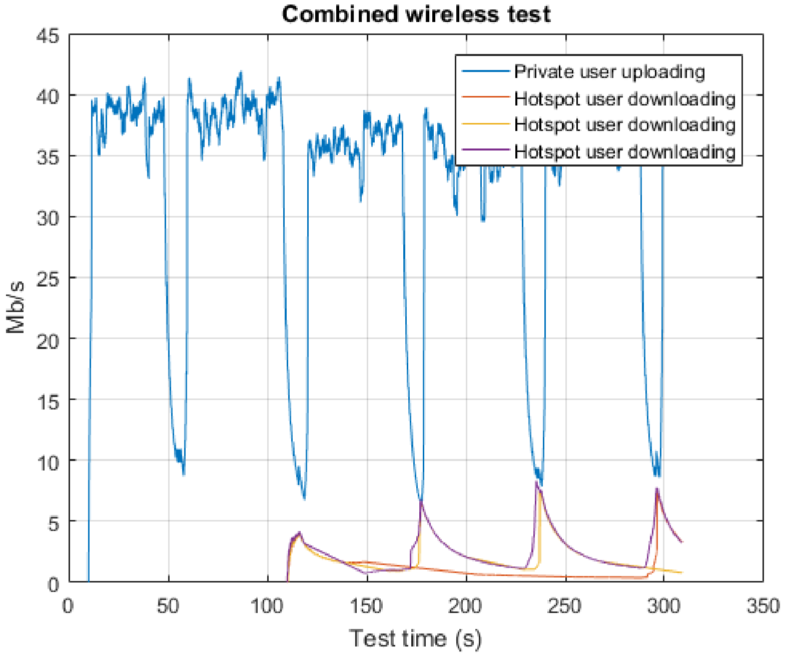


Figure 12.8: A private user starts uploading, followed by three downloading hotspot users delayed by 100 seconds.

12.4 Results Summary

This Section will summarize the previous test conclusions in order to make the results easy understandable. If the wireless medium is congested, the following is concluded:

- *X* The private users wireless download bandwidth gets degraded if hotspot users are uploading. See Figure 12.6.
- *X* The private users wireless upload bandwidth gets degraded if hotspot users are uploading. See Figure 12.3.
- ✓ The private users wireless upload bandwidth does not get degraded if hotspot users are downloading. See Figure 12.8.
- *X* The private users wireless download bandwidth does not get degraded if hotspot users are downloading. See Figure 12.5.
- ✓ The hotspot users are sharing the wireless download and upload resources. See Figure 12.5 and 12.3.
- ✓ The hotspot users can in some cases not utilise the whole spare download capacity on the wireless medium due to a very low priority See Figure 12.5

It is very difficult to ensure QoS on the wireless medium because the WiFi protocol uses probabilities of transmission. So if the two networks transmits over the same channel, one can only provide private users with the largest share of it. The next chapter will discuss and conclude on these findings, and what it means of the overall implementation of QoS and how likely the wireless medium will become a bottleneck.

CHAPTER

THIRTEEN

CONCLUSION

13.1 Conclusion & Discussion

From the tests performed in previous Chapters it is concluded that Ethernet and/or IP QoS can be implemented to favour traffic from private users on the xDSL link. However, the wireless medium can potentially congest in a way that would reduce the private users upload capacity. The potential risk increases with higher capacity on the xDSL link, and decreases with higher capacity on the medium. It was shown that hotspot users can in fact impact the throughput of the private user when they compete for the wireless medium in the upload case. This implies that it could be necessary to apply QoS on all connected mobile devices, which is a tedious task that requires extra software on the connected devices. This extra software should, in theory, be able to use the QoS capabilities of the WMM technology to change traffic priorities generated by a device. Another approach could be upgrading the hardware and software of the MAG to add a second SSID on a different wireless channel. In the case of a second channel, the two networks would not have to share the wireless medium and it will thus, indirectly, ensure wireless QoS.

At the moment the xDSL link will always have a lower capacity than the wireless medium. In this case, the issue with congestion on the wireless medium only applies when devices uses the UDP protocol. UDP does not have any congestion avoidance and will thus not care about dropped packets in the MAG, this will keep the device transmitting as fast as possible. However, TCP uses a network congestion-avoidance algorithm that in the case of a dropped packet, will lower the transmission rate. This implies that in the case of congestion on the xDSL link, the MAG starts dropping hotspot traffic, and all hotspot devices will thereby lower their transmission rate. This prevents the TCP protocol from congesting the wireless medium and allows the wireless capacity to be available for private users. However, if Telenor increases their capacity of the xDSL link to support a higher capacity than the wireless medium, the TCP protocol will congest the medium just as much as the UDP protocol.

Since the risks in upload QoS relies on the amount of UDP traffic generated, Telenor has provided some traffic statistics in order to proper assess how likely this will become an issue. They state that UDP, in upload peak periods, accounts for $2.65\,\mathrm{Gbps}$. This is approximately a third of all upload traffic measured in bytes, with almost no difference between mobile and xDSL customers. It means that UDP traffic is in fact a large portion of the overall traffic, so it is possible that the wireless medium become a bottleneck, depending on the services using UDP on the mobile devices. The extend of the issue should be tested thoroughly by Telenor, before the hotspot solution is launched nation wide.

The implemented download solution in the MAG works as desired, except in the case, where the xDSL link provides a capacity higher than approximately $15\,\mathrm{Mbps}$. In Figure 12.5 it was shown that hotspot users can not download faster than $15\,\mathrm{Mbps}$, implying that hotspot users might not able to utilise the entire spare capacity. This is caused by the low priority assigned to hotspot users on the radio, which lowers their probability of transmission.

An interesting point to mention concerning the download implementation is UDP fragmentation. When UDP packets are fragmented, only the first fragmentation includes the UDP header with the destination port, which means that it is only possible to apply download QoS on the first fragmentation of the UDP packet [notes [2016]]. The rest of the packets will not have a destination port, and the BRAS would thus not know to which network it belongs. Telenor has not been able to provide any good statistics on how common fragmented UDP packets are, but from their traffic statistics they state that only 7% of all download data during peak hours is UDP traffic.

The following list is a repetition of the requirements shown in Chapter 9. The list shows which requirements are approved and which requirements are not approved by the configuration and network tests:

Note: Requirements marked green are approved, red are disapproved, and orange are more or less approved.

Must have

1. Current upload QoS implementations will remain intact.
2. Current download QoS implementations will remain intact.
3. Hotspot users upload traffic will not decrease the xDSL subscribers download or upload resources.
4. Hotspot users download traffic will not decrease the xDSL subscribers download or upload resources.
5. No extra software is required on hotspot user devices and private user devices.

Should have

6. Hotspot users will evenly share the available upload resources.
7. Hotspot users will evenly share the available download resources.
8. The internal resources of the MAG should always favor private users.

From the list it is seen that some requirements 4 to 8 is approved, while requirements 1 to 3 is not completely approved. It has been chosen to put requirements 1 and 2 in the *more or less approved* category, because the *default* IP QoS label's was modified in Listing 10.3 in Section 10.1.1. It is changed from *4* to *6* for both **defclass** and **ackclass**, in order to put traffic from private users in another IP QoS queue than traffic from hotspot users. The configuration should not have any impact on other traffic queues, but as the default QoS implementations has been changed, requirements 1 and 2 can not be fully approved. Requirement 3 is disapproved as upload traffic from hotspot users can impact both the upload and download traffic of private users.

All in all, it can be concluded that QoS can be guaranteed for download traffic in the cases where extensive UDP fragmentation does not occur and in the cases that the bandwidth[1] of the WiFi connection is higher than the bandwidth of the xDSL connection. QoS can only be guaranteed for upload traffic if hotspot users is using the TCP protocol, and it can not be guaranteed if hotspot users are using the UDP protocol as it will not impact the wireless medium. The use of the UDP protocol in upload traffic is however only an issue for high capacity xDSL links, as everyone is competing on the wireless medium on equal term, and as IP QoS/xDSL link will always favor traffic from the private user.

In order to ensure QoS on the wireless medium, the authors propose two possibilities:

1. Telenor can develop an application required to connect to the hotspot network, that would give hotspot users a lower probability to transmit on the wireless medium.
2. Telenor can invest in a new CPE with capabilities of using two separate wireless channels at the same time.

The first proposed solution is based on probabilities and can not completely ensure QoS. The second proposed solution requires Telenor to invest in new hardware, but gives them the possibility to always ensure QoS, if it can be guaranteed that the two networks will never use the same channel.

[1] The QoS is not only restricted to the bandwidth, but is also influenced by latency, jitter, etc.

13.2 Further Work with QoS

Before the solution is ready, the QoS implementation discussed in Chapter 11 needs to be implemented on the BRAS. When the setup has been configured on the BRAS, further tests needs to be performed to ensure that the BRAS does in fact ensure download QoS.

Chapter 12 made it clear that QoS can not be guaranteed on the wireless medium. It is clear that there are no QoS for the upstream data path, and the downstream data path is based on transmit probabilities. During the tests, all wireless clients were very close to the access point and were stationary. It could be relevant and interesting to see how hotspot users and private users throughput changes with distance to the AP, as well with obstacles blocking line of sight. Will the results be the same?

Furthermore, in Section 11.1 the solution only implements port translation for MAGs that uses the PPPoE protocol. If the PPPoA protocol is to be used, additional NAPT templates should be implemented. It is possible to implement two identical port translations, one for each interface.

Further research into how likely the wireless medium is to be congested would be advised, since the proposed solution can not guarantee upload QoS in such cases.

A combined list of the CLI command configurations of the MAG, shown in this Part, can be found in Appendix H Section H.2 on page 242. The command list have been modified slightly to save the amount of printed pages. The full list of unmodified configurations can be found in the *ListOfCLICommands.txt* file on the group server: `http://kom.aau.dk/group/16gr1022/`. The next Part will look into Authentication and User Identification in order to make sure that only Telenor mobile subscribers can connect to the hotspot solution, and to make sure that Telenor can follow the laws of data retention.

Part III

Authentication and User Identification

CHAPTER FOURTEEN

INTRODUCTION & ANALYSIS

Part III will focus on both authentication and user identification, as the topics are quite similar. In this project user identification is defined as the ability to intercept traffic and relate that traffic to a specific mobile subscriber. A requirement for the hotspot project is that only Telenor mobile subscribers will be able to connect to the hotspot solution and thereby freely benefit from all the advantages the solution offers.

Due to law regulations, Telenor is required to be able to track and, if necessary, store all user activities on the Internet. By tracking traffic generated by each hotspot user, it prevents the situation where the xDSL subscriber is accused of actions performed by mobile subscribers. However, this obligation makes the task very difficult as it include a set of complex challenges that can be define as follows:

User Identification is the ability to identify and recognize individual users connected to the hotspot network and intercept their traffic. User identification consists of the following subparts:

- **Traffic Distinguishing** is the capability to distinguish traffic generated by the private users or from individual hotspot users.
- **Traffic Interception** is the way user traffic is intercepted.
- **User Information** is the ability to gather necessary information of individual users connected to the hotspot network.
- **Combiner** is the part that combines **User Information** with **Traffic Interception** in order to finalize the **User Identification**. The **Combiner** could e.g. be a system that stores collected information where it belongs.

As the focus of the project is on the MAG and the BRAS has been inaccessible during the project, the solution will focus on the **User Information** and **Traffic Distinguishing** subparts.

The aim of this Chapter is to give an introduction to enterprise level authentication and accounting methods. Section 14.1 includes a short description of WPA-Enterprise authentication protocols, popular AAA servers, and an insight to the authentication and accounting processes. Section 14.2 will give thoughts on how to distinguish traffic from individual users in a private network. Part III also includes Chapter 15 that presents a solution design for authentication and user identification, and Chapters 16 and 17 implements and tests each configuration individually.

14.1 Enterprise Level Authentication and Accounting

When Telenor mobile subscribers wishe to connect to the hotspot network they will need to be authenticated to ensure that only Telenor mobile subscribers can use the hotspot solution. All Telenor mobile subscribers should be able to connect to all MAGs running the hotspot solution; There are four different possibilities to authenticate access to the solution:

1. All Telenor mobile subscribers will get a Pre-Shared Key (PSK) that is the same for all MAGs belonging to Telenor xDSL subscribers.
2. The MAG will have a hard-coded table of all Telenor mobile subscribers who have permission to connect to the hotspot.
3. The MAG will have access to an online database to see if a user is allowed to connect.
4. The MAG will forward a connect request to an authentication server which will determine if a host will have their request approved.

Of the four options, number four seems the most reasonable choice as a PSK *(1)* could easily fall into the wrong hands. A list of allowed users will include sensitive user data that the public should not be able to access by tampering with the MAG *(2)* or by acquiring access to the online database *(3)*. Option *(4)* is commonly used in large enterprise networks and offers greater security and scalability with a centralized network access. The MAG itself does not know about user permissions and the authentication table is typically inaccessible through the Internet[1]. As there are various authentication protocols and different authentication servers, some of the widely used will be presented in the next Section.

[1] Everyone MAG can access the authentication server, but all data between the mobile subscriber and authentication server is encrypted.

14.1.1 WPA-Enterprise Authentication Protocols

An introduction of *Wireless security protocols* were presented in Part I, Section 3.4.1. This Section will introduce authentication protocols for *WPA-Enterprise* networks, which uses an authentication server. **WPA-Enterprise** is an 802.11x IEEE authentication standard, based on the Extensible Authentication Protocol (EAP), used in enterprise environments that require the implementation of an authentication server. EAP is not only limited to wireless networks, but can also be used with wired networks and point-to-point connections for the price of a more complicated setup, compared to the WPA-PSK method. By using EAP there is a gain in scalability and additional security against some types of attacks. There are various EAP authentication protocols for WPA-Enterprise networks, each with benefits and drawbacks, and some of the most common authentication protocols are:

EAP-Transport Layer Security (TLS) is an IETF standard using Transport Layer Security and is the most common and secure EAP standard used. It requires a certificate on both the authentication server and the client, which makes it secure. Even if an intruder acquires user name and password, the intruder will still need the user certificate.

EAP-Protected Extensible Authentication Protocol (PEAP) is the second most used EAP standard with a digital certificate only on the server. A secure TLS tunnel is established between the client and server. The server authenticates itself to the client with the certificate, and the user credentials are then exchanged.

EAP-Subscriber Identity Module (SIM) is used to authenticate SIM users. When a client requests a connection, a 128-bit challenge is sent from the authentication server to which only the right SIM card can answer. This method offers great security and whilst is it very easy to use as it does not require any user interaction as the information needed is taken from the SIM card.

As there are a large range of different authentication servers that use the 802.11x authentication standard, the most commonly used will be presented in the next Section.

14.1.2 Authentication, Authorization, and Accounting (AAA) Servers

An authentication server usually does not come alone and in many cases the Authentication, Authorization, and Accounting (AAA) protocol is handled by a single server. The AAA protocol consists of three different parts which are as follows:

Authentication offers the possibility to identify and authenticate users who wishes to connect to a network. The user will usually have to enter valid credentials such as username and password which is compared to an authentication table. By means of comparison the server will determine if the user will be rejected or granted access to the network.

Authorization determines how connected users possess different levels of authority. When an user is authenticated the authorization part checks the user's permissions for various activities, resources, and/or services.

Accounting keeps logs of user activity and statistics. It might be measurements of used resources like data usage or tracking the time a user is connected to a network. The logs can e.g. be used for billing or auditing purposes.

Some of the most widely used AAA servers for 802.11x are the following:

RADIUS: Remote Authentication Dial-In User Service is a client/server protocol of the application layer that uses a connectionless User Datagram Protocol (UDP) session. It is frequently used in enterprise and ISP networks to handle authentications for both wired and wireless connected clients. One of the most widely used and deployed RADIUS servers is the free and open source **FreeRADIUS**. FreeRADIUS is very stable, scalable and supported by a large number of vendors. It is easily modified and deployed for a large range of different security systems. All these advantages has convinced some of the largest companies, ISPs and academic communities to use it.

TACACS+: Terminal Access Controller Access-Control System Plus is a Cisco property protocol that uses a Transmission Control Protocol (TCP) session, instead of the connectionless UDP session used by RADIUS. By using a TCP session, TACACS+ is e.g. more resilient to server crashes as it is possible to determine if a server is running or not. All information sent between server and client is encrypted and TACACS+ it is thereby resistant to many RADIUS vulnerabilities. AT the moment TACACS+ exist as IETF draft proposal [D. Carrel [1997]] and is a newer version of publicly available TACACS protocol [Carl Rigney [1993]].

Diameter: Supports all RADIUS features with some enhancements like error notification and the mandatory use of TCP or Scalable TCP (STCP). It is expected that Diameter will replace RADIUS in the future[Victor Fajardo [2012]].

It is decided that the authentication server, used in this project, will be the **FreeRADIUS** RADIUS server. The choice is made because FreeRADIUS is a widespread free software used by many organisations. FreeRADIUS is easy to install and works almost out-of-the-box. It is easy get access to documentation pages and find setup guides for various system implementations. Another reason is that Telenor already uses RADIUS servers to handle CPE and mobile subscriber authentications and that the MAG only supports RADIUS servers. The next Sections will look into the authentication and accounting processes.

For a connection to a AAA server to work, it is necessary to specify an AAA server IP address, a secret AAA server key, an AAA server port for client authentication (*default port:* `8012`) as well as an AAA server accounting port (*default port:* `8013`).

14.1.3 The Authentication Process

As mentioned earlier, RADIUS uses a client/server model where a Network Access Server (NAS) [Mitton [2000a]][Mitton [2000b]] (or EAP terminology - Authenticator [Bernard Aboba [2003]]) acts as RADIUS client. The NAS is a gateway to the authentication server for clients in the network, that receives user requests and forwards their credentials to the RADIUS server. The RADIUS server, based on the credentials received from the NAS, validates each user and responses the NAS with the necessary user information. The NAS acts accordingly to the response from the server and allows or denies access to particular resources. In the case of the project, the role of the NAS falls to the MAG, which will be responsible for forwarding appropriate user information to the specified RADIUS server.

The RADIUS standard [Carl Rigney [2000b]] defines and specifies the process of authentication and authorization. It outlines elements like operations and messages used for information exchange. In addition, a RADIUS extension [Bernard Aboba [2003]] introduces the support for the EAP protocol to be used with the RADIUS standard. Based on these standards, a detailed process of authentication can be described. Firstly the authenticating user starts the communication with the NAS in order to access desirable resources. The NAS negotiates the use of the EAP protocol and sends an **EAP-Request** message to the user. It is done in order to identify the user, who will respond with an **EAP-Response** message. Afterwards the NAS divides the response into smaller chunks, called EAP-Message attributes, and forwards them to the RADIUS server, as the RADIUS cannot interpret the **EAP-Response** as it is. The first message in the process of authentication and authorization is called the **Access-Request** message:

Access-Request is a message sent to the RADIUS server, by the NAS, that requests a connection to be authenticated and authorized. The message contains port on which the user connects, a NAS Identifier, and user credentials. The user credentials are based on the security protocol used e.g. username and password for EAP-PEAP or SIM credentials for EAP-SIM.

As RADIUS uses the UDP protocol, messages can be lost. In case of not receiving a response from the server, the **Access-Request** message is re-sent or it can be sent to another server. When the RADIUS server receives the **Access-Request** message, it compares the information in the message with its own database or an external database. Authorization is not only limited to validation of username and password, but can also be validated by other requirements chosen by the RADIUS server. The server will respond with the following message:

Access-Challenge is a message that indicates that the request fulfills the requirements, but the server can send additional challenges for which the user should properly respond to with another **Access-Request**.

The server creates an **Access-Challenge** response that is sent back to the NAS. The NAS transforms it into a proper EAP message, that is only understood by the user, who has to respond with a proper answer. With the help of the NAS, the message is sent to the server. For the RADIUS server to perform a proper decision, multiple messages are exchanged to gather all necessary information. Based on the validation, the RADIUS server can respond to the NAS with one of the following messages:

Access-Accept is the message that indicates that the connection, or user, has been authenticated and authorized. The NAS provides the necessary resources to the connected client.

Access-Reject means that the request has not passed the validation and access is denied.

The Authentication process is illustrated in Figure 14.1.

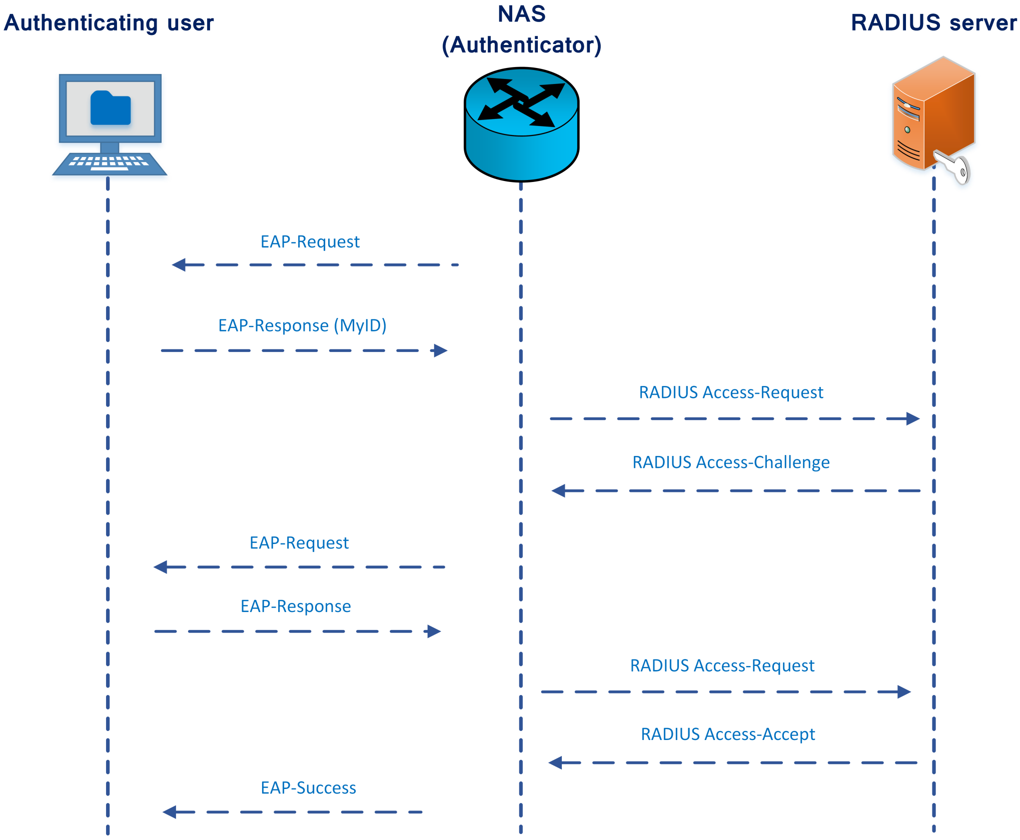


Figure 14.1: The authentication and authorization message flow.

14.1.4 The Accounting Process

The main purpose of the accounting process is to gather and store logs on user resource consumptions, which can be used for billing purposes or future capacity and usage planning. [Rigney [2000]] specifies the details about the accounting operations between the NAS and the RADIUS Server. The accounting process starts after the NAS has received the **Access-Accept** message from the RADIUS Server, and the NAS sends its first accounting message (*Start* message), to the RADIUS server, that indicates the start of a user session.

Accounting-Request is sent by the NAS, to the RADIUS server, with accounting information for the accepted connection.

The **Accounting-Request** message requires an acknowledgment from the server, such that, in case of no response from the server, the message can be resent or sent to an alternative server. The moment the RADIUS server successfully has received the start **Accounting-Request** it sends back the **Accounting-Response** message.

Accounting-Response is sent in response for the **Accounting-Request** message, and indicates the request has been successfully recorded and processed.

During a session, the NAS can send periodic updates (is not required to), that is sent in the form of **Accounting-Request** *Update* messages. To end a session, the NAS will always send a special type **Accounting-Request** called a *Stop* request. The message accounts specific statistics like time or the number of input/output packets. The session usually ends in two ways: The user logs out or the session times out. The accounting process is illustrated in Figure 14.2

RADIUS attributes

A RADIUS message carries specific information about the authentication, authorization and accounting details used between the client and the server. These details are called attributes. Each message contains a header that indicates which type of message it is, and includes zero or more attributes. The message can include information of e.g. **Access-Request**, **Access-Accept** in the authorization process or *Start* and *Stop* messages in the accounting process. Some of the attributes has to be presented in specific messages and some are arbitrary. In the case of **Access-Request**, the message needs to contain attributes with user credentials e.g. username and password for EAP-PEAP. The **Access-Accept** message may contain attributes that specifies the type and constrains of the connection that is allowed for a particular user. The attributes are specified in RFC2865 for authentication [Carl Rigney [2000b]] and RFC2866 for logging [Rigney [2000]]. Vendors has the possibility to design and include Vendor-Specific Attributes (VSA) for their products. These VSA attributes are also used by the xDSL forum in RFC4679 [Vince Mammoliti [2006]], where attributes for the xDSL technology

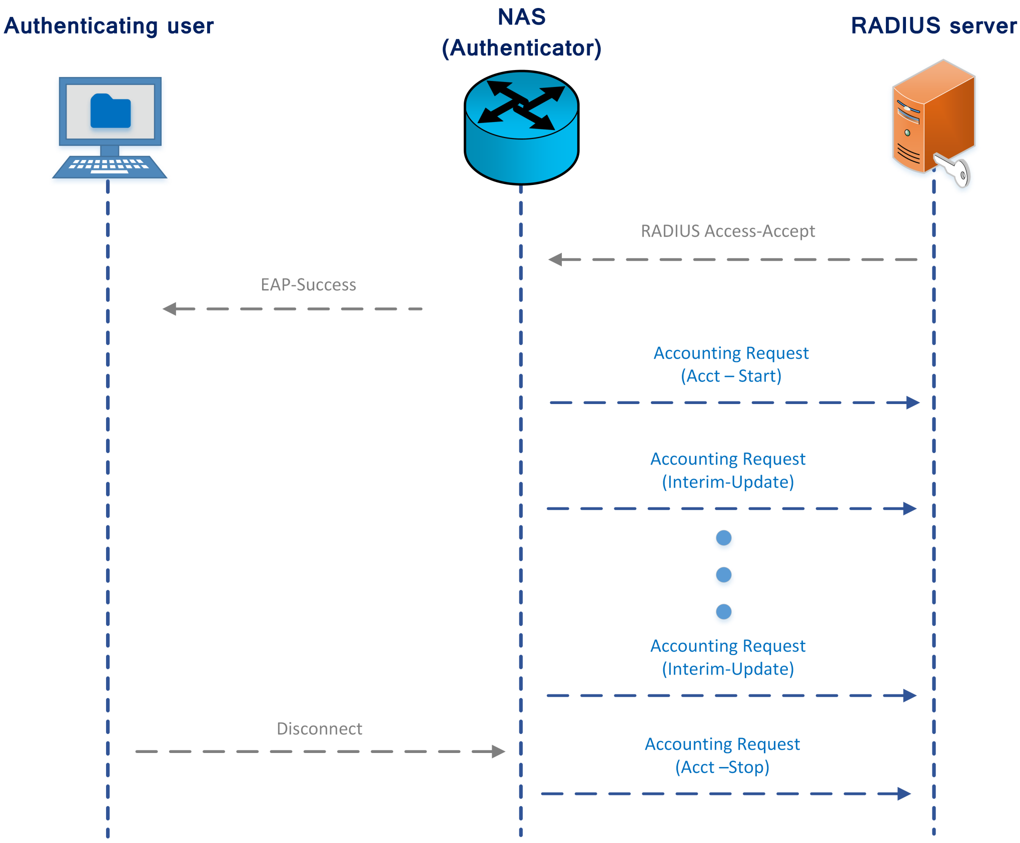


Figure 14.2: The accounting message flow. All the messages in the accounting process requires acknowledgment from the RADIUS server.

are specified. An example of xDSL attributes are *Maximum-Data-Rate-Upstream* and *Maximum-Data-Rate-Downstream*, which can be seen in either the accounting messages or the **Access-Request** messages. Specific VSAs can be sent in order to specify link connection on the BRAS [Juniper Networks [2011]]. Over time, additional RADIUS attributes has been added e.g RADIUS extensions in RFC2869 [Carl Rigney [2000a]] or the support for tunnel protocols in RFC2867 [Glen Zorn [2000]].

The list of possible logging attributes is extensive, but the attributes that occur in this project can be seen in the following list. The not so relevant attributes are written in gray text, while the most relevant attributes, for the project, is given in black text:

Acct-Session-Id: Is a unique session ID for easier recording of start and stop messages. Identical numbers must be used in all accounting messages. It may also be used in an **Access-Request**, then the NAS has to use the same session ID in the **Accounting-Request**. The attribute is not unique if the NAS has been rebooted.

An identical number has to be used in all accounting messages.

Acct-Status-Type: Defines the **Accounting-Request** type e.g. *Start* or *Stop* message.

Acct-Authentic: Defines how the user was authenticated. It can be by the NAS itself, the RADIUS or any other protocol.

User-Name: Specifies the name of the user that is authenticated.

Framed-IP-Address: Identifies the internal IP address of the user.

NAS-IP-Address: Identifies the IP address of the NAS that, on behalf of the user, is requesting the authentication.

NAS-Identifier: Is a description that specifies the NAS that is requesting the authentication.

NAS-Port: Is the physical port number (See Section 3.2.1 on physical ports). It is not a UDP or TCP port, but is the physical connection on the NAS from which the authentication request is sent.

Called-Station-Id: Allows the NAS to send phone number or MAC address of the station (NAS) that the user called or connected to.

Calling-Station-Id: Is the phone number or MAC address of the call (client).

NAS-Port-Type: Specifies the NAS physical port type (See Section 3.2.1 on physical ports).

Connect-Info: Is sent from the NAS to the RADIUS and identifies the user connection to the NAS.

Event-Timestamp: Records the time of the **Accounting-Request** message occurrence.

Acct-Unique-Session-Id: Is a FreeRADIUS attribute. By mixing, **Acct-Session-Id** and other attributes, creates an ID unique for each connection.

Timestamp: Is a FreeRADIUS timestamp

Acct-Session-Time: Is used in *Stop* **Accounting-Request**, defines how long the user used the services in the session (given in seconds).

Acct-Input-Packets: Defines the number of sent packets, by the user, during a session. Can be used in the **Accounting-Request** *Update* message, and is used in the *Stop* message.

Acct-Output-Packets: Defines the number of received packets, for the user, during a session. Can be used in the **Accounting-Request** *Update* message, and is used in the *Stop* message.

Acct-Input-Octets: Defines number of octets sent, by the user, during a session. Can be used in the **Accounting-Request** *Update* message, and is used in the *Stop* message.

Acct-Output-Octets: Defines the number of octets received, for the user, during a session. Can be used in the **Accounting-Request** *Update* message, and is used in the *Stop* message.

Acct-Terminate-Cause: Defines why the session was ended. It could be by an error caused by the NAS or by the RADIUS server, or can be caused by a user request and many more. Is used in the **Accounting-Request** *Stop* message.

If the VSA attributes are used, both the NAS and RADIUS server has to support them. If the NAS sends unknown attributes to the RADIUS server, they will be ignored or can cause an error by e.g. switching the result with another attribute. It is a common problem that the NAS does not support required attributes, and as not all NASs supports the same attributes, it is important to examine the NAS vendor specification for supported attributes [van der Walt [2011]]. As for the project, Technicolor has not specified which attributes is supported by the MAG. However, by experience, it is known that the attributes in the list above are supported by the MAG.

14.2 Distinguishing Traffic

The ability to distinguishing the traffic from each user is needed in order to fulfill the law regulations, that ISPs in Denmark, needs to obey. The concept of distinguishing traffic was introduced for download QoS in Part II (Sections 8.2 and 9.2 and Chapter 11). In order to do proper bandwidth allocation between private users and hotspot users, the traffic originating from each network needs to be identified. However, in the case of user identification the problem is more complex, as the goal is not only to identify traffic from each network, but to identify traffic generated by each hotspot user in the hotspot network. A few possible solutions have been taken into consideration for solving the issue, which is very related to Section 8.2 in Part II:

QoS tag: A QoS tag can be added in order to identify traffic generated by each hotspot user. The tag can be added to either the layer 2 frame or the layer 3 IPv4 header. As presented in Section 8.2.1, the solution requires Telenor to invest in new hardware, which makes the solution very expensive.

PPP tunnel: A unique PPP session can be created for each internal hotspot IP address allocated in the hotspot network as explained in Section 8.2.2.

Port translation: As explained in Section 8.2.3, port translation can be used to distinguish traffic from each hotspot user, if traffic from each hotspot user is mapped into a fixed port range. The idea is simply to let all traffic from the private network be mapped into a specific port range allocated for the private network, and let traffic from each hotspot user be mapped into its own port range.

Multiple public IP addresses: Lastly, as described in Section 8.2.4, a possible solution is to assign a public IPv4 address for each internal hotspot IP address. The concept is to assign one public IP address for all private users and then allocate an individual public IP address for each internal hotspot IP address. When a hotspot user connects to the MAG it will be possible to identify all traffic belonging to that specific user by the public IP address assigned.

An introduction has been given to the enterprise level authentication and accounting methods as well as thoughts to how it is possible to distinguish traffic generated by different hotspot users. The next Chapter will, by the analysis given in this Chapter, come up with a solution design that solves authentication and user identification for the project.

CHAPTER

FIFTEEN

SOLUTION DESIGN

This Chapter will provide a solution design for both authentication and user identification. It is based on the requirements and the analysis given in previous Chapter. The requirements for the Authentication and User Identification are as follows:

15.1 Requirements

Authentication

The main goal of Authentication is to make sure that only Telenor mobile subscribers can use the hotspot solution. The requirements are as follows:

Must have

1. Only Telenor mobile subscribers are allowed to access the hotspot network.
   (a) Hotspot users should be authenticated through an RADIUS server.
   (b) User are authenticated with the EAP-SIM authentication protocol.
2. No extra software is required on hotspot user devices and private user devices.

Should have

3. User authentication should be seamless for the user.

User Identification

The main goal of User Identification is to make sure that the xDSL subscriber is protected against the consequences of illegal activities performed by hotspot users. This is done by following the laws of data retention. As the scope of the project has been limited to only look into the **User Information** and **Traffic Distinguishing** subparts, the requirements are as follows:

Must have

1. The solution will be able to distinguish traffic generated by different hotspot users.
2. The solution should give Telenor the necessary information to do traffic interception for each hotspot user.
3. Each hotspot user connected to the hotspot network can be identified.
4. The solution will give Telenor the necessary information to combine user information and intercepted traffic.
5. No extra software is required on hotspot user devices and private user devices.

15.2 System design for Authentication

Section 14.1.1 presented three commonly used EAP authentication methods for WPA-Enterprise networks, and Section 14.1.2 presented the some of the most commonly used AAA servers. As the MAG only supports RADIUS servers it was chosen to use the FreeRADIUS RADIUS server. For ISP networks, It is very common that the authentication server is reachable from the BRAS, such that all MAGs are able to send login requests. The proposed system will thereby follow the design shown in Figure 15.1.

The idea is that the FreeRADIUS server will use EAP-SIM WPA-Enterprise authentication, as all hotspot users will be Telenor mobile subscribers with a SIM-card. As the hotspot users are subscribers of Telenor, Telenor has access to the needed SIM information to create the 128-bit authentication challenge. A good thing with EAP-SIM authentication is that the user does not have to input any username or password, which allows for a seamless connection.

15.2.1 Limitation: EAP-PEAP in place of EAP-SIM

EAP-SIM authentication requires access to user sensitive data, that can not be accessed directly without clearance from Telenor[1]. The difference between EAP-PEAP and

[1] EAP-SIM authentication can not be done without a SIM card and needs an AAA server that has information about this exact SIM-card such that a *challenge* can be sent.

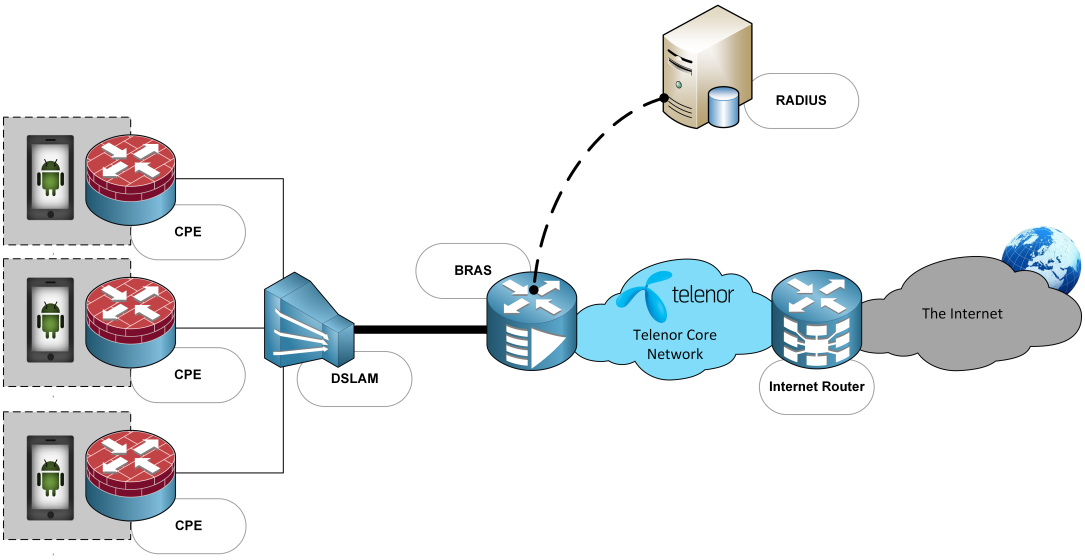


Figure 15.1: Proposed system design for user authentication with the RADIUS server reachable by the BRAS. The phones connected to the CPEs are the hotspot users that gonna be authenticated.

EAP-SIM is somewhat narrow and it is thereby chosen to make a RADIUS server using FreeRADIUS that uses EAP-PEAP authentication. The benefit from this choice is that no time is wasted on acquiring the clearance, which might not improve the project much, as the design choices for different EAP authentication protocols is somewhat equal.

15.3 System design for User Identification

This Section describes the solution design and choices made in order to fulfill the requirements for user identification described in section 15.1. Several methods that can be used to identify users as presented in 14.2. Three of the methods possesses few, but very clear disadvantages, which was also described in Section 9.2.

QoS tag: As mentioned, the QoS tag requires Telenor to invest in new hardware. The solution will thereby become very expensive, but will also become very complex to implement and manage.

PPP tunnel: Establishing an unique PPP session for each hotspot user requires resource allocation for each tunnel. Many of Telenor's xDSL subscribers are connected to a TDC DSLAM so that the solution might become expensive, as Telenor pays TDC per PPP session. Moreover, modules in the BRAS can support only finite number of ppp sessions thus introducing additional hundreds of thousands of new ppp sessions will require additional hardware.

Port translation: The port translation method might seem complex to manage, but with a combination of NAPT rules and session accounting, the solution is very cheap and requires a minimal amount of extra hardware (if any). It terms of implementation costs, complexity and expenses, port translation may have clear advantages over the other methods.

Multiple public IP addresses: Introducing a new public IPv4 address for each hotspot user is very costly in terms of resources (i.e. IPv4 addresses). Even though Telenor possess a large pool of unused IPv4 addresses, this is not a good way to spend the resources. The hotspot solution is at the moment designed to serve up to eighth hotspot users, and it requires Telenor to allocate an additional IPv4 address for each hotspot allocation on each xDSL subscription. With 150.000 xDSL costumers as mentioned in 1, it results in around 1.2 million additional IPv4 addresses that needs to be allocated.

As seen, port translation might has clear advantages over the other methods. It has already been implemented for Download QoS in Chapter 11, and it might be better to add a bit of complexity to an already existing solution, than creating additional implementations. In the system proposed, **User Identification** system actually consists of a combination of several subsystems as described in the beginning of Chapter 14. Only two subsystems of the four subsystems will be implemented in this project: A **User Information** system, that is used to relate a specific user in the hotspot network to an authenticated mobile subscription, and a *Traffic Distinguishing* system, that is used to distinguish traffic from hotspot users. For the full system to work a combination of NAPT rules in the MAG, traffic interception at the BRAS, and session accounting at the MAG and at the RADIUS server, must be made in order to track and store traffic generated by each hotspot user as shown in Figure 15.2. In the Figure the green part of the system will be made in this project, and the red part is out of the scope of this project[2], as the combination of user information and traffic interception can be done it various ways, depending on what and how much information Telenor wishes to collect.

If the port range already reserved for hotspot users is split into equal size chucks, it should be possible to allocate a specific port range for each connected hotspot user, and thereby distinguish their traffic. A fixed number of ports, with combination of an accounting server, can lead to an easy and automatic management of traffic inception. It is also an universal solution, as ports is a layer 4 entity, which means that it can be used with different layer 2 and layer 3 technologies, making it robust to changes in the network architecture.

In Chapter 11 the port range 5000-26847 was defined for hotspot traffic. If the range issplit into 8 equal size chucks, it should still be large enough to meet all the demands of a user connected to a hotspot network. Each user will obtain a range of 2731 ports which is more than enough for smart devices like smartphones, tablets or laptops. The

[2] As mentioned in Section 9.2, the BRAS has been inaccessible during the project, and a system at the BRAS will not be introduced. It is known that the BRAS is able to do traffic interception. How that traffic is forwarded further into the system has not been investigated.

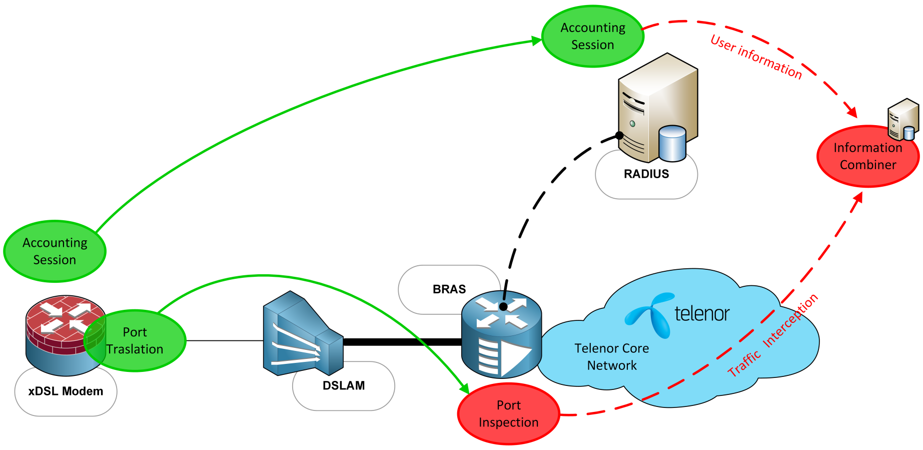


Figure 15.2: A combination of NAPT rules, traffic identification, and session accounting is used to track and store traffic generated by by each hotspot user. The green parts will be made in this project, and the red parts is out of the scope of this project.

port ranges allocated should be fixed, such that the first usable IP address in the hotspot network will be assigned the first 2731 usable ports. The solution is presented in Figure 15.3, and together with Table 15.1 the dedicated port range for each user is defined. When the FreeRADIUS server is configured to support accounting, the combination of a fixed port range as well as accounting information, from the FreeRADIUS server, should enable a clear and easy way to identify each of the users and to fulfill required law regulations.

User	IP address	Dedicated port range
1	192.168.13.37	5000-7730
2	192.168.13.38	7731-10461
3	192.168.13.39	10462-13192
4	192.168.13.40	13193-15923
5	192.168.13.41	15924-18654
6	192.168.13.42	18655-21385
7	192.168.13.43	21386-24116
8	192.168.13.44	24117-26847

Table 15.1: Dedicated port ranges for hotspot users

A solution design has now been presented for authentication and user identification. Chapter 16 will implement and test the proposed design for authentication, and Chapter 17 will implement and test the user identification solution.

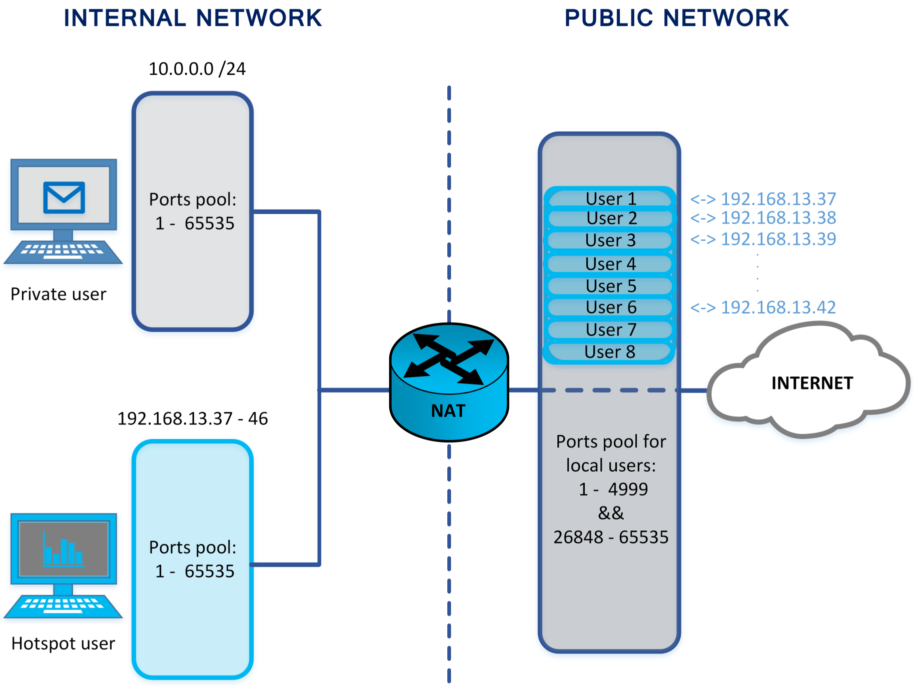


Figure 15.3: Hotspot port translation

CHAPTER

SIXTEEN

IMPLEMENTATION AND TEST FOR AUTHENTICATION

This Chapter will present the configuration of the FreeRADIUS RADIUS server and test if the server is able to correctly approve/deny hosts, who are connecting using the EAP-PEAP authentication method. The main focus of the Chapter is to test the FreeRADIUS RADIUS server and to check if the MAG supports the FreeRADIUS RADIUS server. The configuration of FreeRADIUS is done in Kali Linux, which is a Linux distribution based on Debian. To configure FreeRADIUS a step-by-step guide made by the YouTube account *ethical hacker* is followed [ethical hacker [2013a]][ethical hacker [2013b]]. The guide is easy to follow and it is not practical to reinvent the wheel as the main focus is not the configuration itself.

16.1 FreeRADIUS Configuration

The Kali Linux computer is located and connected to the AAU network as illustrated in Figure 16.1. The AAU network has been configured by *IT Services* such that the Kali Linux computer has a public IP i.e. `192.38.55.78`, which has port-forwarding defined for the port range `20000-20099` (`20012` chosen as authentication server port in */etc/services*. See Listing 17.6 on page 177). The public IP of the MAG is `85.81.240.13`, and the MAG is added as a client with RADIUS secret *"radiusSecret"* in the *clients.conf* file[1] as seen in Listing 16.1 on the following page.

[1] Unless specified all configuration files are accessed from folder */usr/local/etc/raddb*

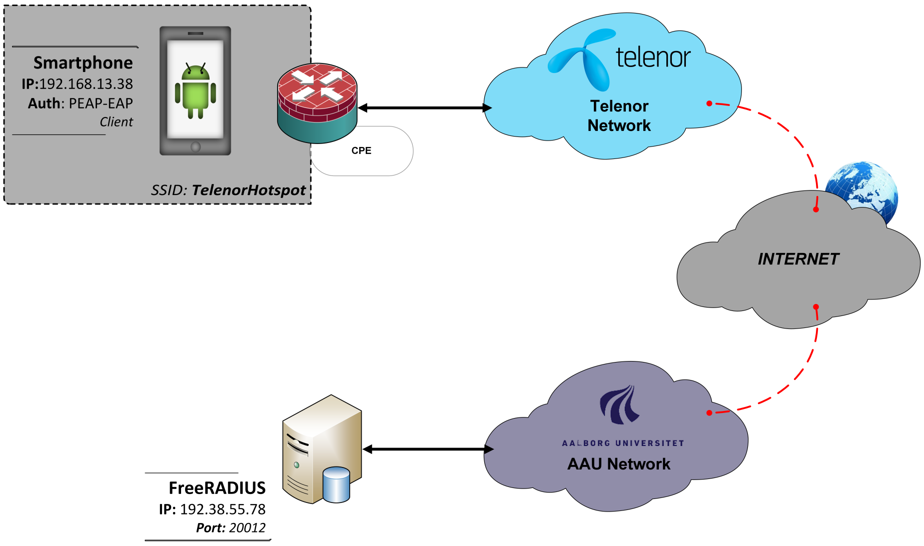


Figure 16.1: The MAG is connected to the Internet through Telenor's core network. The RADIUS server has a public IP at the AAU network, that can be accessed on port 20012.

```
[...]
client TelenorRADIUS {
    ipaddr       = 85.81.240.13
    secret       = radiusSecret
}
[...]
```

Listing 16.1: Configuration of a client (i.e. the MAG) for the RADIUS server in *clients.conf*.

The EAP-PEAP protocol requires users to have matching username and password, and two users are thereby added in the *users.conf* file as seen in Listing 16.2. Both users can connect to the SSID **TelenorHotspot** with the EAP-PEAP authentication protocol. FreeRADIUS has been configured to use EAP-PEAP in the *mods-available/eap* file.

```
[...]
UserOne Cleartext-Password := "SecurePassword"
UserTwo Cleartext-Password := "12345678"
[...]
```

Listing 16.2: Configuration of the two users who are allowed to connect to the **TelenorHotspot** SSID. The configuration is done in *users.conf*.

The configuration for FreeRADIUS is now finished. Next Section will test if allowed hosts can connect, while hosts with incorrect username and/or password are denied.

16.2 FreeRADIUS Test

The MAG supports the use of RADIUS servers for authentication and in Part I, Section 5.1.1 the MAG is configured with the following settings seen from Listing 5.2:

SSID: TelenorHotspot

Security mode: WPA-Enterprise

WPA version: WPA2

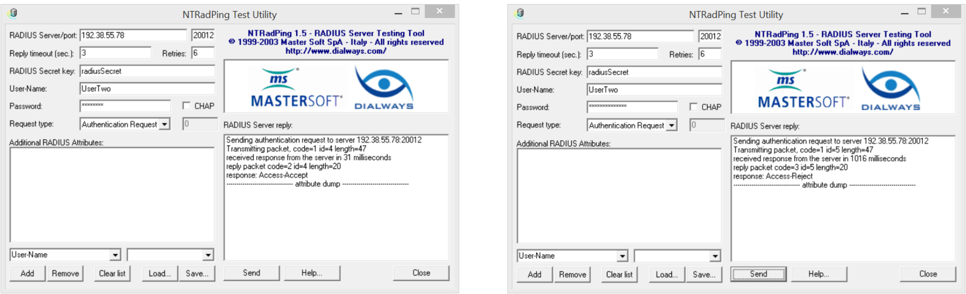
RADIUS IP: `192.38.55.78`

RADIUS port: `20012`

RADIUS key: "radiusSecret"

That configuration gives the MAG the necessary information to connect to the FreeRADIUS RADIUS server.

To test the setup, the FreeRADIUS server is started in debugging-mode in the *Terminal* with the command **radiusd -X**. By using different devices it is concluded that the MAG works with the FreeRADIUS RADIUS server, and users are able to authenticate themselves. To better test the limits of the setup, a test is performed with *NTRadPing*, which is a tool to test RADIUS servers in Windows [Geier [2011]]. *NTRedPing* is easy to use and only requires the IP address, port number and RADIUS secret of the authentication server. Figure 16.2 shows the tool in use. It is seen that **UserTwo** is *accepted* in the left picture and **rejected** in the right picture as a wrong password is used.



NTRadPing Test Utility

RADIUS Server/port: 192.38.55.78 20012
Reply timeout (sec.): 3 Retries: 6
RADIUS Secret key: radiusSecret
User-Name: UserTwo
Password: ********
Request type: Authentication Request 0

RADIUS Server reply:

```
Sending authentication request to server 192.38.55.78:20012
Transmitting packet, code=1 id=4 length=47
received response from the server in 31 milliseconds
reply packet code=2 id=4 length=20
response: Access-Accept
------------------------------ attribute dump ------------------------------
```

NTRadPing Test Utility

RADIUS Server/port: 192.38.55.78 20012
Reply timeout (sec.): 3 Retries: 6
RADIUS Secret key: radiusSecret
User-Name: UserTwo
Password: **************
Request type: Authentication Request 0

RADIUS Server reply:

```
Sending authentication request to server 192.38.55.78:20012
Transmitting packet, code=1 id=5 length=47
received response from the server in 1016 milliseconds
reply packet code=3 id=5 length=20
response: Access-Reject
------------------------------ attribute dump ------------------------------
```

NTRadPing 1.5 - RADIUS Server Testing Tool
© 1999-2003 Master Soft SpA - Italy - All rights reserved
http://www.dialways.com/
MASTERSOFT
DIALWAYS
□ CHAP
Additional RADIUS Attributes:
User-Name
Add
Remove
Clear list
Load...
Save...
Send
Help...
Close

Figure 16.2: The Figure shows the tool *NTRadPing* is used to test the FreeRADIUS server. *Left Figure* shows **UserTwo** who signs in with **12345678** and is thereby *accepted*. *Right Figure* shows **UserTwo** who signs in with **SecurePassword** and is thereby *rejected*.

Table 16.1 shows the result of the test. It is seen that the FreeRADIUS RADIUS server performs as expected. Users with matching password (defined in Listing 16.2) are accepted and users with an incorrect match are rejected.

A FreeRADIUS RADIUS server has been implemented. Users can authenticate using the EAP-PEAP authentication protocol and the MAG as well as FreeRADIUS

Username	**Password**	**Password**	**Password**
UserOne	🟩 SecurePassword	🟥 12345678	🟥 *SecretPassword*
UserTwo	🟥 SecurePassword	🟩 12345678	🟥 *SecretPassword*
UserThree	🟥 SecurePassword	🟥 12345678	🟥 *SecretPassword*

Table 16.1: A test is done with three different users and three different passwords. Red=Rejected, Green=Accepted.

works as expected. Users with wrong username/password match are rejected while a correct match accepts authentication requests. As it has now been confirmed that the MAG supports the RADIUS setup, the next Chapter will implement and test user identification.

CHAPTER

SEVENTEEN

IMPLEMENTATION AND TEST FOR USER IDENTIFICATION

As described in Section 15.3, it is chosen to use a combination of port translation and an accounting server to collect information of hotspot user generated traffic. Section 17.1 will present the configuration needed for doing port translation in the MAG, to make sure that each usable IPv4 address in the hotspot network has a fixed port range, such that traffic can be intercepted at the BRAS. As the BRAS has been inaccessible during the project, a simple test will be performed in order to confirm the accuracy of the implemented port translation solution. Moreover, Section 17.2 will do additional configuration of the MAG and the FreeRADIUS RADIUS server to support session accounting to relate hotspot users to a specific mobile subscription. The solution is tested to show how user information is stored during an accounting session.

17.1 Port Translation

When an user initiates a connection to the MAG, the device is given an IP address by the DHCP server as described in Sections 3.3.3 and 5.1.4. A way to identify a specific hotspot user, is to do port translation, and then inspect that translated port at the BRAS. To do this hotspot users must be given a unique range of ports, depending on which IP address they are assigned. The concept of port translation was described in Sections 8.2.3 and 11.1, and it is chosen that NAPT templates are the best choice compared to NAPT mapping. In the MAG the port that will be translated is named **inside_port** and is used within the private network whereas **outside_port** is the port that is used after the port translation and together with the public IP, this is the port visible to the Internet. Port translation in the MAG allows only for mapping port ranges in a one-to-one manner, as described in Section 8.2.3. It means that an equal size of **inside_port** ports and **outside_port** ports must be used when creating NAPT rule templates. The downside is that it is not possible to map all (1-65535) possible ports to a range of *2731* ports with a single template.

Figure 17.1 illustrates how the port mapping is done for one hotspot user. The first usable IPv4 address in the hotspot network is `192.168.13.37` (**inside_addr**) and the dedicated **outside_port** port range for that address is 5000-7730 as shown in Table 15.1 on page 163. Because of the "one-to-one rule", it is necessary to split the port range 1-65535 into 24 small chunks of **inside_port** port ranges, into the size of one **outside_port** port range, which is 2731 ports. The 24 **inside_port** port ranges is shown in Table 17.1.

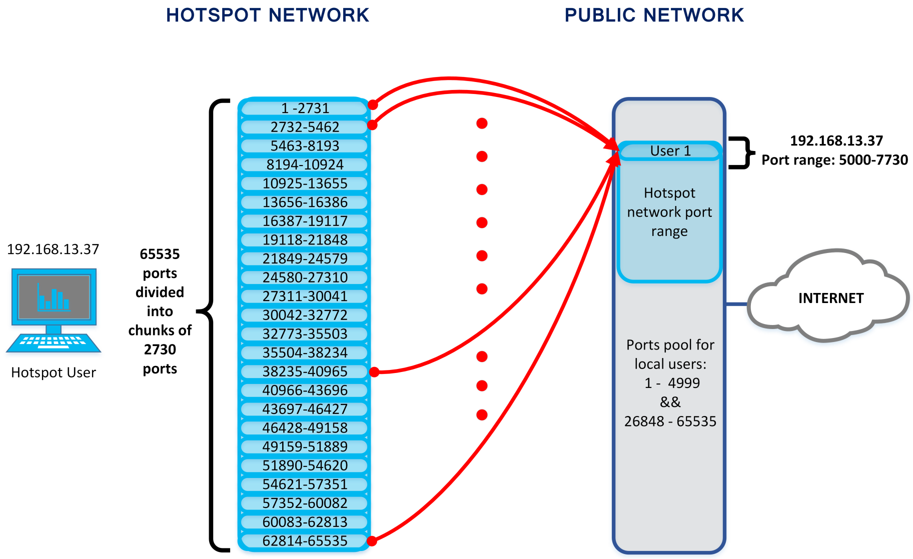


Figure 17.1: Port mapping for one hotspot user. It is seen that all ports, in small chucks, are mapped into a small port range.

Listing 17.1 shows a "*default*" template for NAPT rules. All traffic that is routed externally goes through the xDSL link and uses either one of the *internet_pppoe* or *internet_pppoa* interfaces. In this case it is chosen to make the NAPT mapping template for the *internet_pppoe* interface, as the link used for this particular xDSL modem uses PPPoE. The **outside_addr** is specified to be `0.0.0.1`, as it points to the *first* public IP[1]. X is a variable that specifies the IP address of the hotspot user, which in this design can have the values *37-44*. The variable Y specifies the number of the rule. The variables wwwww, and yyyyy specifies the dedicated **inside_port** port range for each rule also according to Table 17.1. The variables zzzzz, qqqqq is used to specify the dedicated **outside_port** port range for each hotspot user according to Table 15.1 on page 163.

[1] One xDSL modem can have multiple public IP addresses, and `0.0.0.1` is used as a *wildcard* for the first defined public IP address. `0.0.0.2` is used for the second public IP address (if any), and so on. The wildcard is used as public IP addresses can be assigned dynamically to an xDSL modem [Technicolor [2008]].

Rule	Dedicated port range (**inside_port**)
1	1-2731
2	2732-5462
3	5463-8193
4	8194-10924
5	10925-13655
6	13656-16386
7	16387-19117
8	19118-21848
9	21849-24579
10	24580-27310
11	27311-30041
12	30042-32772
13	32773-35503
14	35504-38234
15	38235-40965
16	40966-43696
17	43697-46427
18	46428-49158
19	49159-51889
20	51890-54620
21	54621-57351
22	57352-60082
23	60083-62813
24	62814-65535

Table 17.1: **inside_port** port range for each rule.

```
=> nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=192.168.13.X outside_port=zzzzz-qqqqq inside_port=
    wwwww-yyyyy mode=outbound weight=10 status=up description=User
    XRuleY
```

Listing 17.1: Port mapping for an arbitrary user in the hotspot network.

In order to show how the NAPT rules can be generated, Listing 17.2 is shown. The Listing shows how MATLAB can be used to generate the commands necessary to configure the rules. The following definitions has been made: *numusers=8*, *numrules=24*, and the *outsiderange* and *insiderange* matrixes are defined according to Tables H.1 and H.2.

```
numusers = 8;
numrules = 24;

insiderange = [1,2731; 2732,5462; 5463,8193; 8194,10924;
    10925,13655; 13656,16386; 16387,19117; 19118,21848;
    21849,24579; 24580,27310; 27311,30041; 30042,32772;
    32773,35503; 35504,38234; 38235,40965; 40966,43696;
    43697,46427; 46428,49158; 49159,51889; 51890,54620;
    54621,57351; 57352,60082; 60083,62813; 62814,65535];

outsiderange = [5000,7730; 7731,10461; 10462,13192; 13193,15923;
    15924,18654; 18655,21385; 21386,24116; 24117,26847];

for user = 1:numusers
    for rule = 1:numrules
            str = sprintf('nat tmpladd intf=internet_pppoe type=
                napt outside_addr=0.0.0.1 inside_addr=192.168.13.%d
                 outside_port=%d-%d inside_port=%d-%d mode=outbound
                 weight=10 status=up description=User%dRule%d',(
                user+36),outsiderange(user,1),outsiderange(user,2),
                insiderange(rule,1),insiderange(rule,2),(user+36),
                rule);
            disp(str)
    end
end
```

Listing 17.2: An extracted script from a piece of MATLAB code that creates NAPT rules.

As NAPT by default translates everything into the dynamic port range (49152-65535), no NAPT rule templates will be added for private users.

It should be noted that a high number of NAPT rules is required to translate ports for each IP address. The maximum limit of NAPT rules in the MAG is 256, with 30 rules already allocated by default, which restrains the number of possible usable IP addresses for the hotspot network. In the setup, 24 rules has been created for each hotspot user, which limits the maximum amount of hotspot users to 9. It is chosen to use 8 users as it leaves some space for new rules while it also leaves some available ports. As the highest possible port to be assigned to hotspot users is port 49151 and there is only space for 226 NAPT rules, the problem is an optimization problem that requires a trade-off between the number of rules, the maximum port number, and the amount of hotspot users wished as illustrated in Figures 17.2A-C. Figure 17.2C gives the combination of the two restrictions, where the orange dot is the point chosen with 8 users and 24 rules per user, and the black dot is the optimal point with 12 users and 18 rules per user. Technicolor, the MAG manufacturer, could perhaps change the software such that NAPT rules could easier be implemented for solutions like this.

Now, when all the rules have been explained and configured the next Section will examine if the port translation works as expected in practise.

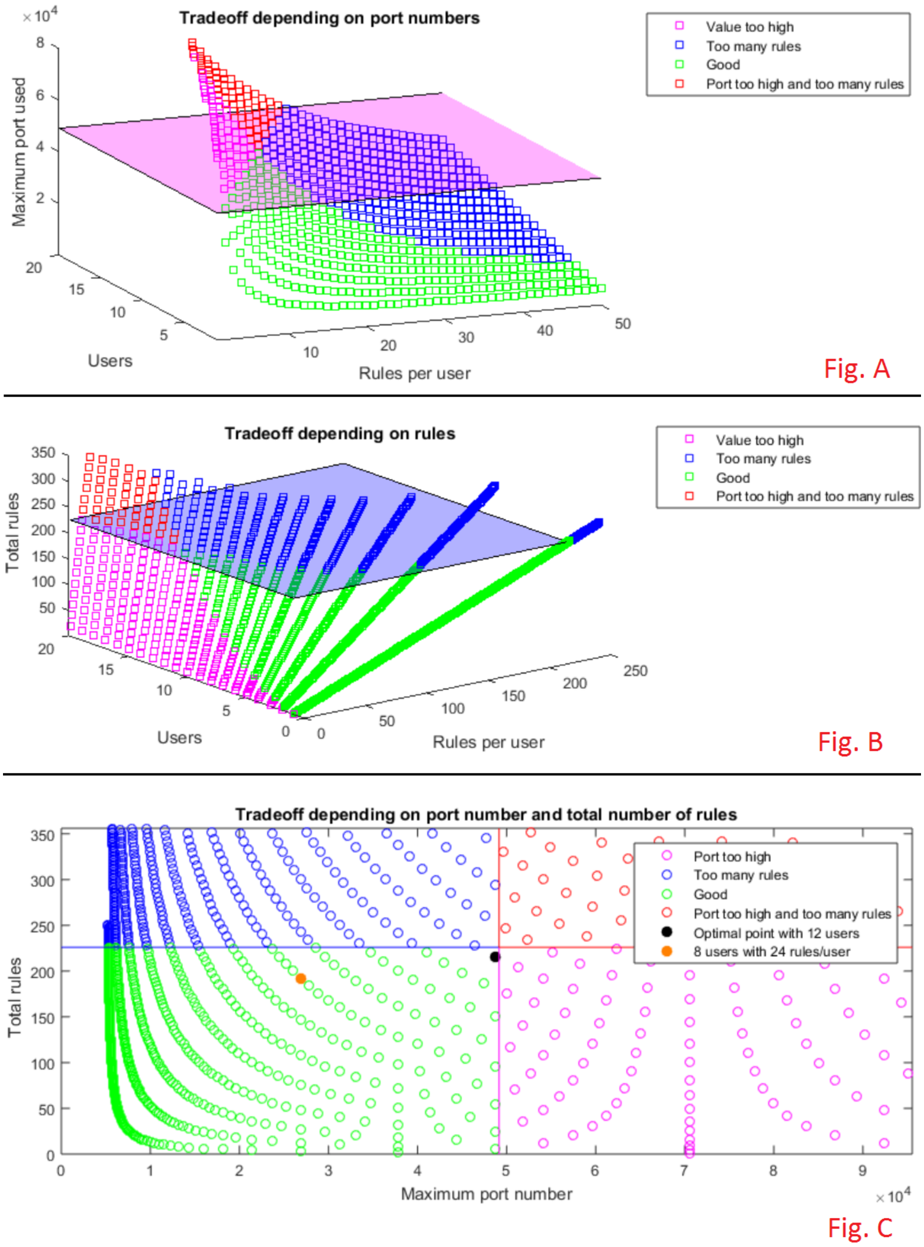


Figure 17.2: The optimization problem to find the maximum amount of hotspot users. Fig. A shows the tradeoff depending on the maximum port used, and Fig. B shows the tradeoff with the number of rules. Each tradeoff is made between the total number of rules (i.e. 226) and the maximum port possible (i.e. 49151). The port allocation starts at port number 5000. Fig. C shows a combination of each restriction. The maximum number of users is 12 with 18 rules per user as shown with a black dot. The orange dot shows the point chosen with 8 users and 24 rules per user.

17.1.1 Examination of Port Translation

A test of port translation is performed in order to inspect if the port translation configuration done in the MAG performs as expected. Figure 17.3 illustrates the test setup. All eighth hotspot users has been connected to the hotspot network to examine the behaviour of all possible hotspot users. In addition, a few users from the private network has been connected to make sure the default behaviour of private users has not been changed after the implementation of the NAPT rule templates. Each of the users, in turn, establishes a connection with the server on the AAU network. The source port of each connection has been randomly chosen in the range of 1-65535. When the server receives some data, the source port mapped by NAPT is stored. A comparison of the randomly chosen port and the translated port is shown in Figure 17.4. Data in green represents ports from the private users, while other colours shows each of the eight hotspot users. The X-axis is the original source port from the user device, and the Y-Axis shows the source port of the received packet on the server. From the results it can be concluded that the port translation solution works as expected. All ports translated, for each user, falls within the dedicated port range as defined in the Table 15.1 on page 163.

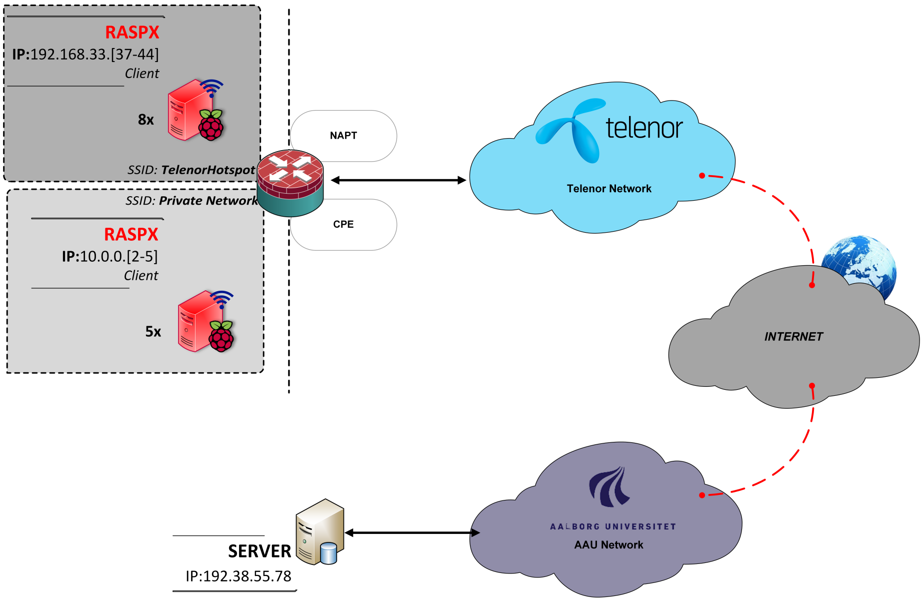


Figure 17.3: The MAG, that does port translation, is connected to the Internet through Telenor's network. The server has a public IP at the AAU.

In Section 11.1, port translation were defined for hotspot users, in order to do download QoS. As all hotspot users the still within the port range 5000-26847, no extra configuration of download QoS needs to be implemented.

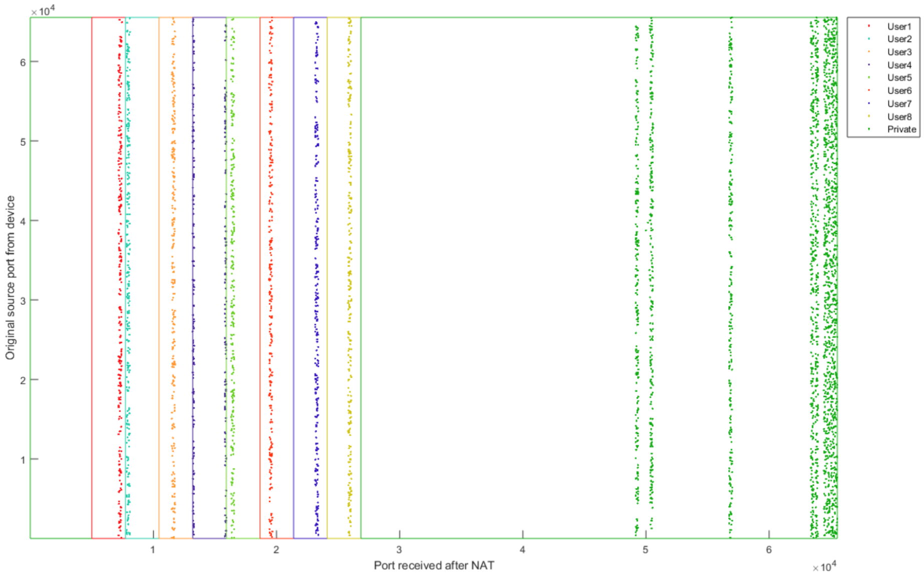


Figure 17.4: Port Translation results. The X-axis is the original source port from the user device, and the Y-Axis shows the source port of the received packet on the server.

17.2 Accounting Configuration

It is assumed that the hotspot network is already configured in the MAG according to Chapter 16, so that the configuration of the MAG already includes the support for the FreeRADIUS server.

By issuing the CLI command => *wireless radius server config*, a list of configured and available RADIUS servers is obtained. Listing 17.3 shows the result of the command. It can be seen that two RADIUS servers are available for each of the configured SSIDs, either an authentication or an accounting server. At this moment the authentication server, for the hotspot network, points to the FreeRADIUS server, created for the purpose of this project, while the accounting server has not been configured.

```
=> wireless radius server config
RADIUS Server Flags: [E]nabled    [D]ead   [A]ctive   DH[C]P
                     [S]witchover [H]ead   [F]QDN
Radius Server Configuration
-----------------------------------
RID [0] SSID [0] No[0] Type [AUTH] IP:port [0.0.0.0:1812] next [1]
    flag [....H..]
RID [0] SSID [0] No[0] Type [ACCT] IP:port [0.0.0.0:1813] next [1]
    flag [....H..]
RID [0] SSID [1] No[0] Type [AUTH] IP:port [192.38.55.78:20012]
    next [1] flag [E.A.H..]
RID [0] SSID [1] No[0] Type [ACCT] IP:port [0.0.0.0:1813] next [1]
    flag [....H..]
```

Listing 17.3: RADIUS servers in the MAG.

In order to enable accounting for the already configured FreeRADIUS RADIUS server, appropriate settings has to be configured as seen in listing 17.4. The RADIUS server (IP: `192.38.55.78`), set up for the purpose of this project, will now act both as an authentication and an accounting server. However, different destination ports has been specified, such that traffic can be distinguished between the two processes - authentication uses port `20012` and accounting uses port `20013`. *type=accounting* specifies that the server will be used for accounting purposes, and by enabling *dhcpaccounting* it allows one to send more attributes that are learned by the DHCP e.g. **Calling-Station-ID**, **Framed-IP-Address**, and **Acct-Terminate-Cause**. The DHCP will also begin generating *Start* and *Stop* messages the moment an IP address is assigned for an hotspot user, or when the lease is terminated by the user or DHCP server.

```
=> wireless radius server config radio_id=0 ssid_id=1 radius_id=0
    state=enabled ip=192.38.55.78 port=20013 dhcpaccounting=yes
    type=accounting secret=radiusSecret
```

Listing 17.4: Enabling accounting on the FreeRADIUS RADIUS server

Listing 17.5 shows the result of the configuration.

```
=>:wireless radius server config
RADIUS Server Flags: [E]nabled    [D]ead   [A]ctive   DH[C]P
                     [S]witchover [H]ead   [F]QDN

[...]

RID [0] SSID [1] No[0] Type [AUTH] IP:port [192.38.55.78:20012]
    next [1] flag [E.A.H..]
RID [0] SSID [1] No[0] Type [ACCT] IP:port [192.38.55.78:20013]
    next [1] flag [E.A.H.C]
```

Listing 17.5: RADIUS servers in the MAG after enabling accounting

In situations where the accounting server is not needed, there is the possibility to change the type of *ACCT* to *AUTH*. This option can be used to point to another IP address with a secondary authentication server.

Accounting runs by default with FreeRADIUS, and no setup of accounting is need. However, as only a small port range (20000-20099) is open on the public IP address assigned by *IT services* at AAU, the default port for accounting purposes is changed in */etc/services*. The port for accounting has been chosen to be `20013`, which is one of the available port on the public IP address `192.38.55.78`.Listing 17.6 shows a fragment of */etc/services* with the newly specified ports for authentication (Section 16.1) and accounting.

```
[...]
radius    20012/tcp               # Radius Authentication
radius    20012/udp               # Radius Acuthentication
radius-acct 20013/tcp radacct   # Radius Accounting
radius-acct 20013/udp radacct     # Radius Accounting
[...]
```

Listing 17.6: Default port configuration is changed in */etc/services*.

17.2.1 Accounting Test

The test setup is similar to the one that was shown in Figure 16.1 in Chapter 16, that was dealing with authentication. Figure 17.5 shows the test setup for the accounting test, where a Samsung smartphone has been used. The user connects to the *TelenorHotSpot* SSID by using the EAP-PEAP method, and disconnects after a while. The accounting log is stored in */var/log/freeradius/radacct*, on the computer running the FreeRADIUS RADIUS server.

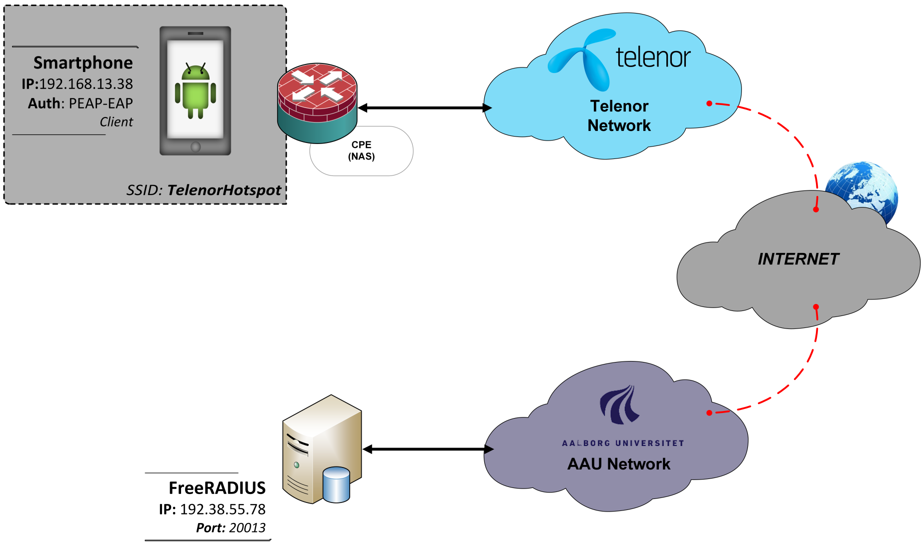


Figure 17.5: The user is connected to the hotspot network using a smartphone. The FreeRADIUS RADIUS server has a public IP at the AAU network. The accounting service is accessed on port 20013.

The first event that occurs, when the RADIUS starts, is that the NAS sends an accounting message indicating that it is ready to send accounting streams to the RADIUS server. The message can be found in the accounting log and is shown in Listing 17.7. The accounting message of type *Accounting-On* states that the NAS, from now on, will perform accounting for its users. The log includes MAC address of the NAS, more precisely a xDSL interface MAC address, as seen in the **NAS-Identifier** attribute.

Moreover, a MAC address of an interface in the NAS is sent in the **Called-Station-Id** attribute. This MAC address is the interface for which the accounting will be performed, and is the interface that hotspot users are connected to.

```
Tue Apr 26 12:54:12 2016
  Acct-Status-Type = Accounting-On
  Acct-Authentic = RADIUS
  NAS-Identifier = "Technicolor30-91-8F-94-46-7A"
  Called-Station-Id = "32-91-8F-94-46-7C:TelenorHotspot"
  Acct-Terminate-Cause = NAS-Reboot
  NAS-IP-Address = 85.81.240.13
  Event-Timestamp = "Apr 26 2016 12:54:12 CEST"
  Acct-Unique-Session-Id = "0cd906bcecc991739110404dd4e2cde2"
  Timestamp = 1461668052
```

Listing 17.7: Accounting is ready message.

When a hotspot user is successfully authenticated, the accounting messages are sent to the RADIUS server. Listing 17.8 represents the message specifying the start of accounting for a particular user. This log includes relevant information about the connected user. First of all, the log includes the attribute called *User-Name* which, as the name suggests, is the name of the authenticated user defined in Listing 16.2 on page 166. In this case the user is called *UserTwo*, because of the use of the EAP-PEAP protocol. In the case of using EAP-SIM, the attribute will use specific SIM card credentials. The attribute named *NAS-IP-Address* contains the IP address of the NAS interface. This is the public IP of the MAG so that the ISP is able to see to which specific MAG the hotspot user is connected to. The log also includes a few timestamps e.g. *Event-Timestamp* that e.g. determines the time the user was connected. The log also gives more specific information about the specific user connected e.g. MAC address of the device that is provided in the *Calling-Station-Id* attribute. The MAC address of the phone is `EC-1F-72-9C-10-24`. However, one of the most important and useful attributes is *Framed-IP-Address* that specifies the hotspot user's IP address within the network. Because of the directly specified and dedicated port range given in Table 15.1 on page 163, for each of the possible hotspot IP address, Telenor has knowledge on which ports a given user/internal IP address operates. By performing port inspection on the BRAS, Telenor is capable of executing packet interception of required user traffic.

```
Tue Apr 26 12:54:25 2016
  Acct-Session-Id = "571F6514-00000001"
  Acct-Status-Type =  Start
  Acct-Authentic = RADIUS
  User-Name =  "UserTwo"
  Framed-IP-Address =  "192.168.13.38"
  NAS-IP-Address =  "85.81.240.13"
  NAS-Identifier = "Technicolor30-91-8F-94-46-7A"
  NAS-Port = 0
  Called-Station-Id = "32-91-8F-94-46-7C:TelenorHotspot"
  Calling-Station-Id =  "EC-1F-72-9C-10-24"
  NAS-Port-Type = Wireless-802.11
  Connect-Info = "CONNECT 1Mbps 802.11"
  Event-Timestamp = "Apr 26 2016 12:54:25 CEST"
  Acct-Unique-Session-Id = "e18399faf4052cf6da9d96d90d621b3c"
  Timestamp = 1461668065
```

Listing 17.8: The **Accounting-Request** *Start* message

While the hotspot user is connected to the hotspot network and are utilizing its resources, the NAS sends periodic updates to the RADIUS server, called an *Interim-Update*, as specified in **Acct-Session-Id**. An example of such an update is presented in Listing 17.9. In addition to the *Start* accounting message, each update gives counts on sent and received packets, sent and received octets, as well as the duration of the session. These attributes are distinguished by the blue and red colors at the bottom of the Listing 17.9. The information is usually used for billing purposes, but in this case the attribute values might be used to check if all the necessary packets have been intercepted.

```
Tue Apr 26 12:59:25 2016
  Acct-Session-Id = "571F6514-00000001"
  Acct-Status-Type = Interim-Update
  Acct-Authentic = RADIUS
  User-Name = "UserTwo"
  Framed-IP-Address = 192.168.13.38
  NAS-IP-Address = 85.81.240.13
  NAS-Identifier = "Technicolor30-91-8F-94-46-7A"
  NAS-Port = 0
  Called-Station-Id = "32-91-8F-94-46-7C:TelenorHotspot"
  Calling-Station-Id = "EC-1F-72-9C-10-24"
  NAS-Port-Type = Wireless-802.11
  Connect-Info = "CONNECT 130Mbps 802.11"
  Acct-Session-Time = 300
  Acct-Input-Packets = 617
  Acct-Output-Packets = 685
  Acct-Input-Octets = 76951
  Acct-Output-Octets = 305427
  Event-Timestamp = "Apr 26 2016 15:02:04 CEST"
  Acct-Unique-Session-Id = "e18399faf4052cf6da9d96d90d621b3c"
  Timestamp = 1461668365
```

Listing 17.9: An example of a periodical accounting update.

The **Accounting-Request** *Stop* message indicates that the session has ended and, if needed, the packet interception should end as well. Listing 17.10 shows the last log of the test session. The attribute **Acct-Terminate-Cause** describes the reason for the session termination. From the log, it can be seen that the phone has successfully disconnected from the hotspot network. The log also includes the exact time of termination. The penultimate attribute in Listing 17.10, specifies the unique session identifier that is unique along all sessions in the RADIUS server. The same identifier is seen in all logs belonging to the same session. It is useful when tracking one particular session, when the log file consists of hundreds of such sessions.

```
Tue Apr 26 13:09:51 2016
  Acct-Session-Id = "571F6514-00000001"
  Acct-Status-Type = Stop
  Acct-Authentic = RADIUS
  User-Name = "UserTwo"
  Framed-IP-Address = 192.168.13.38
  NAS-IP-Address = 85.81.240.13
  NAS-Identifier = "Technicolor30-91-8F-94-46-7A"
  NAS-Port = 0
  Called-Station-Id = "32-91-8F-94-46-7C:TelenorHotspot"
  Calling-Station-Id = "EC-1F-72-9C-10-24"
  NAS-Port-Type = Wireless-802.11
  Connect-Info = "CONNECT 130Mbps 802.11"
  Acct-Session-Time = 926
  Acct-Input-Packets = 1173
  Acct-Output-Packets = 1320
  Acct-Input-Octets = 142887
  Acct-Output-Octets = 519880
  Event-Timestamp = "Apr 26 2016 15:12:30 CEST"
  Acct-Terminate-Cause = User-Request
  Acct-Unique-Session-Id = "e18399faf4052cf6da9d96d90d621b3c"
  Timestamp = 1461668991
```

Listing 17.10: An example of a **Accounting-Request** *Stop* message.

As it can be seen, the logging information gives quite clear visibility into the activities of the internal network. Next, Chapter 18 will present the conclusions of both, the user authentication and user identification issues together with the possible future work opportunities.

CHAPTER

EIGHTEEN

CONCLUSIONS ON AUTHENTICATION AND USER IDENTIFICATION

18.1 Authentication Conclusion

From the requirements and delimitation in Chapter 15, the FreeRADIUS RADIUS server does as required. It was chosen to implement the EAP-PEAP authentication protocol in place of EAP-SIM as the main differences are minuscule. The solution can do WPA-Enterprise authentication through an RADIUS server, which requires a matching username and password. Only users with matching username and password is granted access to the hotspot solution. As it was chosen to use the EAP-PEAP protocol instead of the EAP-SIM protocol, the solution design does not fulfill the overall requirements to the solution as seen in the following list:

Note: Requirements marked green are approved, red are disapproved, and orange are more or less approved.

Must have

1. Only Telenor mobile subscribers are allowed to access the hotspot network.
   (a) Hotspot users should be authenticated through an RADIUS server.
   (b) User are authenticated with the EAP-SIM authentication protocol.
2. No extra software is required on hotspot user devices and private user devices.

Should have

3. User authentication should be seamless for the user.

It has been chosen to more or less approve requirement 1. as the implemented solution is done as a proof-of-concept, which shows that it is possible to use WPA-Enterprise authentication to authenticate hotspot users on with a RADIUS server. In order to fully approve the requirement 1., EAP-SIM authentication must be implemented. Requirement 2. is only more or less approved as user authentication is seamless when the network configuration is set up for the hotspot network. In order to fully approve requirement 2., the user should not do any network configuration at all.

Further work with Authentication

A delimitation were done with regards to the authentication protocol used. It is wished that the authentication should be seamless for Telenor mobile subscribers, who uses the hotspot solution. The authentication should be done with the EAP-SIM protocol, which requires access to user sensitive data. A final setup requires access to the database with SIM information of users, which should provide hotspot users with a seamless connection process.

To let hotspot users authenticate themselves seamless, it is required that they should not do any network configurations or other any other setups. Telenor has provided information that it is possible to *push* network configurations to a mobile device, through the SIM-card. It means that it should be possible to let Telenor mobile subscribers connect seamless to the hotspot solution without them doing any setup beforehand. It should be pointed out that this project has not investigated how to push such configurations to a mobile device.

18.2 User Identification Conclusion

Chapter 14 described that, in order to do **User Identification**, four subsystems is needed i.e. **Traffic Distinguishing**, **Traffic Interception**, **User Information**, and **Combiner**. Of the four subsystems, this project has only considered the **Traffic Distinguishing** and **User Information** subsystems, which are also the most essential. The **Traffic Distinguishing** is implemented by doing port translation in the MAG, and **User Information** is implemented by having accounting logs, with user information on a RADIUS server.

Port translation is implemented in Section 17.1 in order to be able to distinguish traffic at the BRAS. Traffic can be distinguished as dedicated port ranges has been defined for each of internal IP address in the hotspot network[1]. By being able to distinguish traffic, it will ensure that the xDSL subscriber will not be accused of actions done in the hotspot network, if traffic interception is also implemented. Traffic interception has not been implemented at the BRAS in this project as the BRAS has been

[1] As stated, there is a limit to how many NAPT rules that can be assigned in the MAG, which gives a upper limit of 12 hotspot users, who can be connected at once.

inaccessible during the project.

In Section 17.2 it is shown that by utilizing the RADIUS accounting logs it is possible to extract necessary information of hotspot users, who uses the hotspot solution. The accounting log contains information of the physical MAC address of the device connected, the username (SIM-information), IP address of the NAS, and the internal IP address of the hotspot user. This allows to identify each hotspot user and to specify which resources they can utilize. Moreover, with the use of the internal IP address assigned, it is possible to obtain information of which port range a given hotspot user can utilize.

By combining information from the accounting logs, and by doing traffic interception at the BRAS, it should be possible to combine this information with in order to distinguish the activities of each hotspot user.

The result of the requirements given in Chapter 15 is as follows:

Note: Requirements marked green are approved, red are disapproved, and orange are more or less approved.

Must have

1. The solution will be able to distinguish traffic generated by different hotspot users.
2. The solution should give Telenor the necessary information to do traffic interception for each hotspot user.
3. Each hotspot user connected to the hotspot network can be identified.
4. The solution will give Telenor the necessary information to combine user information and intercepted traffic.
5. No extra software is required on hotspot user devices and private user devices.

All requirements has been approved as the implemented solution solves the requirement specification. Requirement 1. is approved as it is possible to distinguish traffic using port translation. Requirement 2. is approved as nessesary information is given to the BRAS, such that traffic can be intercepted for different hotspot users within the internal network. Requirement 3. has been approved as it is possible to gather information of each user, who is/has been connected to the hotspot solution[2]. Requirement 4. is approved at the solution makes it possible to implement the **Combiner** than can combine information from the accounting server with intercepted traffic (See Figure 15.2 on page 163). Finally, requirement 5. is approved as no extra software installations is required.

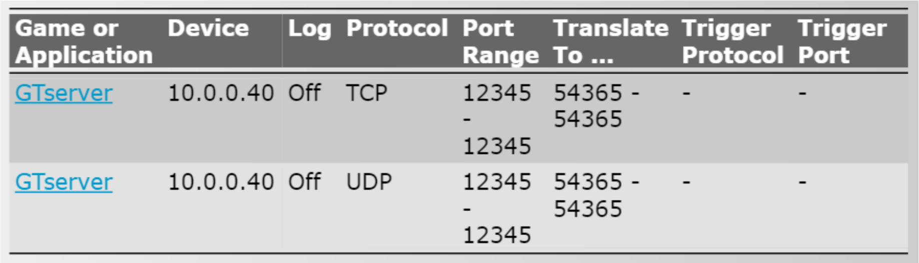
[2] As the EAP-PEAP authentication protocol was implemented in Chapter 16, it is only possible to gather username in form of the username defined in Listing 16.2. If the EAP-SIM authentication protocol was used, this username would be SIM information of the mobile subscriber.

Further work with User Identification

First of all, for the final solution to work, the **Traffic Interception** and **Combiner** subsystems must be implemented. Traffic interception should be done at the BRAS, but not much thought has been given on how further use the intercepted traffic. No matter what, the intercepted traffic must be combined with user information gathered in the accounting logs. A possible solution could be to have a data storage system, that combines user information from the RADIUS server and intercepted traffic from the BRAS. As the RADIUS server provides the accounting information, this can be used to inform the data storage system, which hotspot user uses which ports.

Configuration Change by The xDSL Subscriber

Something that has not been investigated is if the private user chooses to do port-forwarding to e.g. a server. The problem with this port is that the private user will probably pick a randomly chosen port, which can in fact be in the hotspot user range. Lets say that he chooses outside port 12345 for the public IP, that will be mapped to another internal port as can be in Figure 18.1. The problem is that the hotspot user could be accused of something done by the private user if the server has content that should not be on the Internet.



Game or Application	Device	Log	Protocol	Port Range	Translate To ...	Trigger Protocol	Trigger Port
GTserver	10.0.0.40	Off	TCP	12345 - 12345	54365 - 54365	-	-
GTserver	10.0.0.40	Off	UDP	12345 - 12345	54365 - 54365	-	-

Figure 18.1: A private user assigns port 12345 (public IP) to do port-forwarding to port 54365 for the internal IP address 10.0.0.40. The setup if done in the MAG GUI.

Another issue is that the private user can in fact, using his own port translation rules, camouflage all his traffic. It is possible if the private user figures out which hotspot IP addresses and ports are being used. Because of this it is very important to implement some software that denies the private user to do port-forwarding in the hotspot port ranges or to implement his own NAPT rules. It is also recommended to block the private user from seeing which hotspot users are/have been connected to the hotspot solution as mentioned in Section 7.0.4.

UDP Fragmentation

A possible sporadic issue is that the system can not do packet inspection of fragmented UDP packets. As described in Section 13 only the first fragmentation of a UDP packet includes the port, which makes the system not able to classify fragmented UDP packets. In order to investigate if the proposed solution design does not work, an investigation of how many UDP packet are fragmented and what information they typically contain, should be carried out.

PPPoE versus PPPoA

In Section 17.1 it is chosen to only make configurations for MAGs that uses the PPPoE protocol. It is known the some MAGs will use the PPPoA protocol. Additional steps must be taken to take care of these cases, which can be solved in two ways: One, a new, almost identical, configuration method should be created to devices using the PPPoA protocol. This solution requires Telenor to keep track of which MAGs uses the PPPoE and PPPoA protocol. Two, In Figure 17.2 it is shown that it is possible to make NAPT templates for 12 users and still be within the port range limit. A way to implement a solution for both the PPPoA and PPPoE protocols at once, could be to limit the amount of users to 6 (half of the maximum limit of 12), and make a NAPT template for each interface. It means that there will always be two identical port translations, one for each interface. By defining new variables and port ranges, it should be fairly easy to modify the MATLAB script in Listing 17.2 to create new NAPT templates.

The report has focused on the configuration and possibilities of MAG and not an actual implementation in Telenor's network. The proposed solution enables identification of hotspot users and a way to distinguish their traffic. The solution provides the necessary information to later perform packet interception and the ability to combine the information. Telenor has stated that the information given is sufficient to implement a user identification system. A combined list of the CLI command configurations of the MAG, shown in this Part, can be found in Appendix H Section H.3 on page 243. The command list have been modified slightly to save the amount of printed pages. The full list of unmodified configurations can be found in the *ListOfCLICommands.txt* file on the group server: `http://kom.aau.dk/group/16gr1022/`. The next and final Part of the report will give a final conclusion of the project.

Part IV

Project Conclusions

CHAPTER NINETEEN

PROJECT FINALIZATION

This Chapter finalizes the project by summarizing and discussing the work undertaken during the project and by mentioning where solutions requires more work. The project has investigated four different aspects of a hotspot network i.e. Network Integrity, Quality of Service, Authentication, and User Identification, all of which have been grouped into three parts as Authentication and User Identification are much alike. The goal has been to solve the three parts that together cover the overall problem statement from Section 2.1 that states:

"How can one design and implement a hotspot solution for Telenor mobile subscribers on the Technicolor TG788vn v2 modem, that guarantees no loss of QoS for the paying xDSL subscriber, ensures network integrity, and follows the laws of data retention?"

During the project it has been shown that all the parts are highly related to each other and if one part of the system is changed it could easily affect the other parts, meaning that all changes must be taken with care. Section 19.1 will discuss what has been done to solve each part and will end with an overall discussion of the combined solution. The discussion takes its base in the overall systems requirements introduced in Section 2.3. Furthermore, Section 19.2 provides the reader with insight into what must be done in order to finalize the system and which elements require more research due to the limited scope of the project.Finally, Section 19.3 will end the conclusion.

A fully functioning hotspot system solving the problem statement has not been a minor task, it required a wide knowledge of a range of different technologies and procedures.

19.1 Discussion

Three Parts were investigated in order to create a hotspot system that would give Telenor mobile subscribers the possibility of better indoor coverage whilst allowing Telenor to utilise their resources better, by offloading mobile subscribers to wired xDSL connections. The following discussions are based on the three conclusions in Chapters 7, 13, and 18.

19.1.1 Part I: Network Integrity

The goal of Network Integrity is to let hotspot users use the xDSL subscriber's xDSL connection, without introducing additional security issues. From the analysis it was concluded that the best way to ensure network integrity is to create a new subnetwork, for hotspot users only. The solution was tested thoroughly and it is concluded network integrity can be ensured. However, the testing tools used only considered a minor spectrum of possible network security issues, and it is advised to further examine the solution with additional tests. The tests should be performed with special developed software that can e.g. inspect common layer 2 issues. The experiments should be done to make sure hotspot users can not exploit the system or damage the software or hardware. Steps must also be taken in order to make sure that the xDSL subscriber cannot change the configuration of the solution.

19.1.2 Part II: Quality of Service

QoS must be guaranteed as Telenor is obligated to deliver the service that the xDSL subscriber pays for. In an attempt to guarantee QoS, the QoS framework of the MAG and possible network bottlenecks was analysed. It was necessary to examine possible network bottlenecks, as the upstream and downstream data paths might not have the same bottleneck, or they might be treated differently. After QoS was implemented in the MAG, tests concludes that:

Download: It is possible to guarantee download QoS as long as the download bandwidth of the xDSL subscription is below the download bandwidth of the WiFi connection. As QoS can not be performed on fragmented UDP packets, only a scarce amount of or no UDP fragmentation[1] may occur.

As the solution has only been implemented in the MAG, download QoS must be implemented on the BRAS in Telenor's core network in order for a final solution to work.

Upload: Upload QoS can not be guaranteed. It can however, be split into the following cases:

[1] UDP fragmentation is done when a network packet is too large to be transmitted over a connection, and the packet is split into smaller fragments.

- Upload QoS cannot be guaranteed if the UDP protocol are used by hotspot users. The reason for this claim is that the everyone competes for the wireless medium on equal terms, and the UDP protocol does not have procedures to handle network congestion.
- Upload QoS can be guaranteed if hotspot users are using the TCP protocol, as the MAG will drop packets from hotspot users in the case of network congestion at the xDSL link. Hotspot users will notice that the network seems congested and will start transmitting slower and slower. The claim is only true if the upload bandwidth of the xDSL connection is below the upload bandwidth of the WiFi connection, because it requires that hotspot users notices the congestion.

In order to guarantee upload QoS, Telenor must either develop software that gives all hotspot users a lower probability to transmit data on the wireless medium, or they will have to invest in a new CPE that can serve two channels at once, one for each network.

It should be mentioned that it has been proven that upload and download traffic may influence each other in terms of QoS, at least in the wireless medium.

19.1.3 Part III: Authentication and User Identification

The aim of Authentication is to ensure only let Telenor mobile subscribers can use the hotspot solution. An authentication server is implemented that uses the EAP-PEAP protocol to authenticate hotspot users. From implementation it has been proven that the MAG can use RADIUS servers to authenticate clients. For a final system to work the EAP-SIM authentication protocol, that uses the SIM card in the authentication process, should be implemented, as it will give hotspot users the ability to seamlessly connect to the hotspot network.

As Telenor is required to do data retention, User Identification is an important part of the hotspot solution. Different methods to distinguish traffic, and thereby users within a network, have been investigated. It was decided to do port translation, such that each packet includes a specific port that relates to each hotspot user. By using the information, Telenor is able to identify the traffic from and to each hotspot user within the hotspot network. In order to gather information of who uses the hotspot solution, accounting information of each hotspot user is gathered at the RADIUS server whenever a mobile subscriber uses the solution. For a final system to work, traffic should be distinguished and intercepted at the BRAS and combined with the accounting information from the RADIUS server[2]. To fully approve the solution, steps must be taken to make sure that the xDSL subscriber cannot change the configuration of the MAG. Just as with QoS, the impact of UDP fragmentation should be investigated, as fragmented UDP packets cannot be identified by the proposed solution.

[2] The combination of intercepted traffic and accounting information is out of the scope of this project.

19.1.4 Discussion of System Requirements

The project focused on a hotspot solution that took its base in the capabilities of the MAG and Telenor's network. Some important elements for this hotspot solution were inaccessible during the project, but from consultation with the network engineers from Telenor, the choices made in order to solve each part was approved.

In Section 2.3 the system requirements were defined. From the work done, and from the conclusions it is decided to approve and disapprove the requirements as follows:

Note: Requirements marked green are approved, red are disapproved, orange are more or less approved, and blue means that the requirements has not been investigated.

Telenor mobile subscribers

1. Only Telenor mobile subscribers should be able to use the hotspot network.

   Is has been proven that it is possible to implement a solution that only authenticates hotspot users. Even though EAP-SIM has not been implemented, this requirement is approved.

2. Traffic generated by individual hotspot users and the xDSL subscriber should be identifiable.

   From the solution it is possible to identify traffic from each hotspot user. However, fragmented UDP packets cannot be identified, and the requirement cannot be fulfilled.

xDSL subscriber integrity

3. Hotspot users can only utilise the spare capacity of the xDSL connection, so that the xDSL subscriber will not detect the slightest degradation of performance.

   The solution requires a configuration to be implemented in the BRAS. It has been proven that the MAG can handle QoS management, except for wireless upload traffic when hotspot users uses the UDP protocol, meaning that QoS can be guaranteed in some cases, but not always. One must remember that upload can, and will, affect download at the wireless medium. The requirement is not approved as QoS can not always be guaranteed.

4. The network integrity must not be compromised for the the xDSL subscriber as well as mobile subscribers.

   So far, it has been proven that network integrity is not compromised by introducing a hotspot network.

User friendliness

5. Mobile subscribers do not need to do any WiFi network configuration on their device.

   This requirement has not been investigated. However, it is known that Telenor can push network configurations through the SIM card.

6. The solution does not require any extra software installations.

   No extra software installations will be used by either hotspot users or the xDSL subscriber.

7. Mobile subscribers should seamless and *automatically*[3] connect to the hotspot network when in close proximity.

   If WiFi is turned on, smartphones will automatically connect to the hotspot network.

General

8. The hotspot solution will be implemented on the Technicolor TG788vn v2 modem.

   The solution has been implemented on the Technicolor TG788vn v2 modem i.e. the MAG. If another CPE is chosen, all configurations and tests must be made anew.

9. Every hotspot solution should be identical. *In relation to requirement 8., it should be possible to give each xDSL modem the same configuration without specific changes.*

   The solution is configured in a manner that makes it identical if every MAG uses the PPPoE protocol, as used in this project. As described in Section 18.2, a configuration must be created, that gives MAGs, that uses the PPPoA protocol, the possibility to do user identification. The issue is also relevant to ensure download QoS as described in Section 13.2.

10. The hotspot solution will only serve data traffic. Voice calls and text messaging will go through the cellular network.

    The hotspot solution serves only data traffic. No effort has been made to make it ready for voice calls or text messaging.

By combining all parts, it can be concluded that the MAG[4] and this solution have some limitations, which Telenor will have to consider before implementing. The MAG supports multiple subnetworks without introducing any (so far noteworthy) vulnerabilities, it supports the use of a RADIUS server, it can do user identification, but it cannot guarantee QoS. Telenor can not guarantee QoS with the MAG as it is. They can go with a compromise or they can develop an application that gives all hotspot users a

[3] A mobile subscriber should only connect if the mobile device has WiFi turned on.

[4] One of the most noteworthy limitations of the MAG is that it only has one radio, meaning that it uses the same wireless channel for both SSIDs.

lower probability to transmit data on the wireless medium. Another option is to invest in a new CPE, that can serve two different channels at once, one for each network. It is recommended to invest in a new CPE as it is the only solution that can guarantee QoS completely.

If Telenor chooses to implement a hotspot solution, it is recommended that they define a set of service level limitations for different scenarios. A service level limitation could for example be to state the maximum bandwidth Telenor can guarantee xDSL subscribers to always ensure QoS, compared to the restrictions of the wireless medium and so on.

The next Section will comment on further work that must be performed to finalize the hotspot solution as well as commenting on the limitations that were necessary, to limit the scope of the project.

19.2 Further Work

Project delimitations were necessary in order to limit the scope of the project. In Section 2.3 the following project delimitations was given:

> **Handover procedures**: The project will not deal with the handover between cellular networks and the hotspot network. Moreover, it will not deal with the problem of termination of network services when making a handover.
>
> **Data logging**: The project will provide the necessary tools to enable data logging and recording of user information, but will not deal with an implementation at Telenor's core network.
>
> **Only at the CPE**: The project will consider all parts of the system, i.e. the network between the CPE and the Internet, but will only do an actual implementation at the CPE, because of restricted availability to Telenor's core network devices.

It was decided not to deal with **handover procedures** between the hotspot network and the cellular networks, as it is a very complex matter. One of the issues with the proposed system is that the network services on the mobile subscriber's device will be terminated, when the device connects to the hotspot network. It can result in e.g. loss of data or unforeseen termination of essential services. The problem could become very annoying if the mobile subscriber subscriber, simply walking past a hotspot solution, connects because of the close proximity to the WiFi network. The result might be that an important Skype call is terminated, only for being connected to the hotspot network for the five seconds it takes for the mobile subscriber to walk by. A solution to this issue could be a requirement for a piece of software that takes care of the handover problem by allowing hotspot users to keep their cellular network assigned IP address assigned, or by a piece of intelligent software that can decide if the handover should be performed or not.

In Section 15.3 it was decided that the scope of the project is limited with relation to **data logging**. The goal of the project is to only provide Telenor with the necessary information to enable user identification. The reasons for the limitation is because the BRAS has been inaccessible during the project and because of the time that it would require to implement a system that can combine the gathered information. The work required would simply be too much for a single project. The problem is very interesting, and will require knowledge of the structure of cellular core networks and how to do data retention. The reasons here is also why the project worked on a possible solution **only for the MAG**.

A few other interesting aspects that could be investigated are:

1. Will the xDSL subscriber connect to his own WiFi network or the hotspot network, when he is home, if he is also a Telenor mobile subscriber?
2. Can voice over WiFi be implemented for hotspot users?

In order to answer 1. preliminary tests have been performed that indicate networks are picked at random. During the tests, the private network used WPA-PSK authentication and the hotspot network used WPA-Enterprise authentication (EAP-PEAP). The tests were done by making sure that the mobile devices used were not known by any of the networks, and when WiFi was turned on it was discovered that the mobile devices picked each of the networks on equal terms. The tests were performed using Android smartphones. This topic will require more work.

2. has not been given much thought during the project. It should be possible to implement voice over WiFi, but will require one to reconsider the QoS queues of the MAG. A possible solution would be to put hotspot data traffic in queue *1*, xDSL subscriber traffic in queue *2*, hotspot voice over WiFi traffic in queue *3*, xDSL subscriber voice over WiFi traffic in queue *4*, and the xDSL subscriber's VoIP in queue *5*. Each queue will be given different priorities and the higher number means higher priority.

Some of the work done that requires more research, follows below:

Network Integrity:

- More tests must be performed to ensure network integrity. During the project, different network scans were performed, but it might be interesting to undertake advanced attacks using special developed software.

Quality of Service:

- Telenor must decide on what level they wish to ensure QoS for the xDSL subscriber. The current solution can not guarantee QoS and Telenor might have to accept a compromise, as they can not prevent hotspot users from using the UDP protocol. As mentioned Telenor has three possibilities: They can use the MAG as it is, they can develop new software, or they can invest in a new CPE that can serve one dedicated channel for each network.

- QoS must be implemented at the BRAS in order to finish download QoS. The BRAS should identify traffic by doing port inspection.
- It could be relevant and interesting to investigate how hotspot users and private users will act on the wireless medium in a changing environment i.e. they are not stationary.

Authentication and User Identification:

- How to distribute the hotspot network configurations to Telenor mobile subscribers?
- Traffic must be intercepted at the BRAS, and combined with the information from the accounting server, in order to finish User Identification, .

General:

- The xDSL subscriber must be prevented from accessing to some parts of the MAG. By changing the configuration, the xDSL subscriber can get access to information he should not have, he can use the hotspot solution to gain access to hotspot devices or to camouflage his traffic, or he could end up doing harm to himself.
- The solution has only been made for PPPoE. In order to make the solution suitable for PPPoA as well, some configurations must recreated i.e. max associations in Part I and NAPT rules in Part III. *NAPT rules will automatically be changed for Part II from the changes performed in Part III.*
- How common is UDP fragmentation? With regards to data retention, what does it mean when some traffic cannot be identified with the proposed solution? How much will UDP fragmentation affect QoS?

The final Section will conclude on the students' work during the project.

19.3 Project Conclusion

As the project was undertaken in collaboration with Telenor, the most interesting questions to answer must be:

"How can Telenor use the results of the project?"
"What have the students learned/shown?"

The investigations performed throughout the project have shown that the MAG is capable of providing mobile subscribers with a hotspot solution, but only if Telenor defines requirements that clearly state what impact the hotspot solution can have. An example of an impact is: Even though the network integrity is tested thoroughly, there is always the possibility that the hotspot solution might have opened a door that would otherwise have remained closed. As has been shown, the hotspot users affects the QoS, and

Telenor must consider if they are ready to lower the guaranteed bandwidth that they currently claim for some xDSL subscriber, if not, they need to to recognise and take heed of the consequences that might occur. By implementing this solution, Telenor must be ready to respond to unforeseen consequences that might impact on their business image. This project provides Telenor with valuable information on the MAG and identifies possible concerns with the solution. With the information, Telenor should have the ability to determine if it is worth spending time and resources on developing a hotspot solution for the MAG.

The problem of introducing a hotspot network has been very stimulating and rewarding for the. Most of the elements investigated throughout the project were not known to the students at the start of the project, and a lot of knowledge has been gained - including knowledge that has not been documented in the report. Some of the items that made the project very interesting were some of the restrictions and requirements defined by Telenor. A few requirements to mention were: Hotspot users should seamless be able to use the solution without doing any network configurations, no additional software must be installed in order to use the hotspot solution. Both requirements have limited the amount of possible solutions. Another thing to mention is that, even though possible configurations of network core elements have not been investigated throughout the project, the students have always made it clear, with the engineers from Telenor, that each proposed solution should work, and that it is possible to implement the proposed solution at elements not directly described in the report.

An interesting and unexpected aspect that arose during the project was the wondrous bad documentation of the MAG, from Technicolor's side. The documentation of the MAG is at some point lacking very essential information, and the students had, in many cases, to come up with ingenious ways to test restrictions and possibilities of the MAG. In some cases, especially because of the software that is hidden, we had to use our experience and expertise of the MAG to come up with ideas as to how it is designed. To document this claim, Figures 8.3 on page 95 and 17.2 on page 173 is proposed. Each of the Figures is constructed based on information gathered from various tests. The lack of information is also the reason why some parts of the report, related to the MAG, might seem overly well-documented.

From the project work, it should be evident that the students can work and have knowledge/skills within the following areas: Network Analysis, Distributed Systems, Network performance tests, Network Security, Client/Server systems, Wired and wireless networks, Various layers of the OSI model, Quality of Service, Authentication methods, ISP network infrastructure, and problem solving within the area of network engineering. From the report and supervisor meetings, it should be visible that the students are able to communicate scientific problems as well as documenting them.

APPENDIX

A

TELENOR'S XDSL INFRASTRUCTURE AND TOPOLOGY

This appendix aims to give the reader a general overview of the network from the CPE to Telenor's core network. The appendix will not go into details and will only consider the most important parts composing the network, meaning that the information presented will not illustrate the full path in the network.

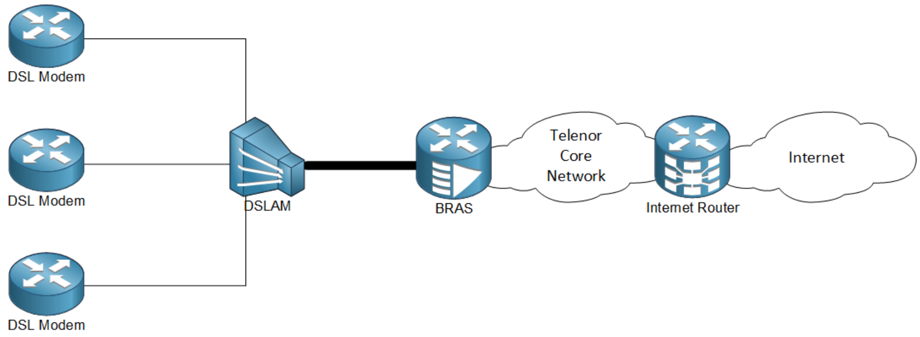


Figure A.1: General overview of Telenor's xDSL network

Figure A.1 presents a simplified physical infrastructure of Telenor's network architecture, that enables the delivery of broadband Internet access for residential subscribers. The architecture consists of a few key elements that are needed to effectively support various services. These key elements are:

xDSL modem is a modem device, occasionally known as the Customer Premises Equipment (CPE), that has a routed path to Telenor's core network. The DSL modem usually acts as both a modem and a router (multidevice) and is located at the residential subscriber. The residential subscriber can usually connect to it through either a Ethernet cable or Wi-Fi.

Digital Subscriber Line Access Multiplexer (DSLAM) is a device that aggregates streams of high-speed data from multiple xDSL customers into a high-capacity link in order to access the ISP network[S. Wadhwa [2011]]. The DSLAM is a Layer 2 device and is seen as the first layer 2 hop in the xDSL instrastructure. It is Virtual Local Area Network (VLAN) aware, but can not route traffic as it acts similar to a switch. Its only purpose is traffic forwarding.

Broadband Remote Access Server (BRAS) is a layer 3 device and is seen as the first layer 3 hop in the xDSL infrastructure. The BRAS intercepts and aggregates traffic from multiple DSLAMs and its main functions are to ensure QoS, do Internet Protocol (IP) management, support VLAN mappings and route traffic through the ISP network to the Internet[Architecture and Group [2006]].The BRAS is also known as a Broadband Network Gateway (BNG), which is the case for Telenor, but is mentioned as a BRAS throughout the report.

To make the description even more precise it should be added that Telenor allows two general services in their network. One service is **data** for ordinary Internet traffic and another is **Voice over IP (VoIP)** that allows for reliable telephone conversations in terms of latency, jitter, and bandwidth. Both services relies on the session control protocol **Point-to-Point (PPP) Protocol**, that is the dominant transport mechanism in ISP networks[Juniper Networks [2014]]. The purpose of a PPP tunnel is to establish a session between two points the in ISP network. Figure A.2 shows two VLANs established between the CPE and a BRAS for the **data** and **VoIP** traffic. PPP tunnel is established inside the **data** VLAN and specify link characteristics e.g speed as well as authenticate the residential CPE.

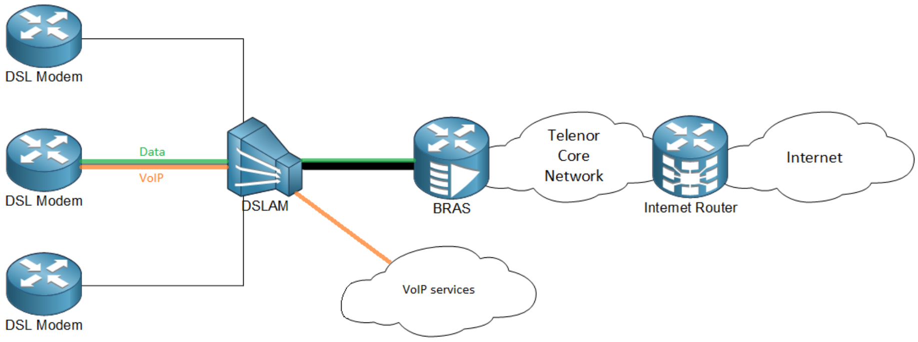


Figure A.2: General overview of Telenor's xDSL network with **data** and **VoIP** VLANs

In Denmark, Telenor is a small ISP with minor cable infrastructure implicating that Telenor is forced to use the physical copper lines of the bigger ISP TDC, which means that most of Telenor's access network connections are rented as they belong to TDC. In a network perspective, it means that most of Telenor's xDSL subscribers are connected to TDC DSLAMs, where connections belonging to Telenor customers are forwarded to a Telenor DSLAM. Usually both DSLAMs, Telenor and TDC, are placed in the same Point of Presence (PoP), which is the aggregation premises of an ISP. Figure A.3 shows how xDSL modems are connected to a TDC DSLAM that is connected to a Telenor DSLAM.

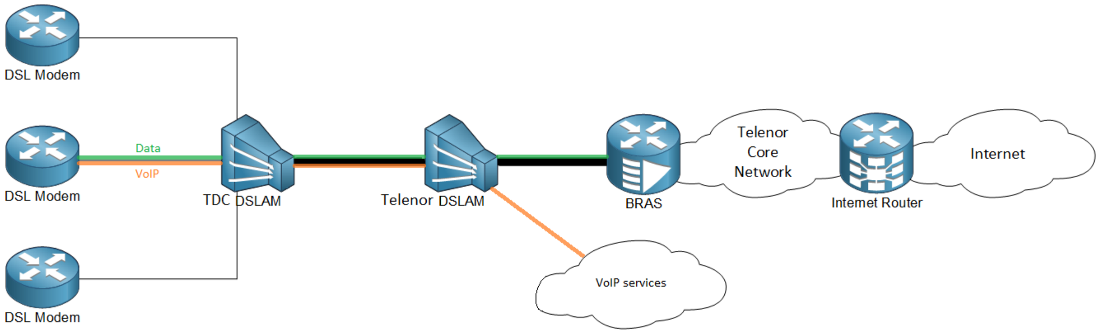


Figure A.3: Overview of Telenor's xDSL network, including TDC DSLAM, with seperate Vlans for Data and VoIP.

The network shown is a very simplified version of the real network, but the main contributing parts for this project has been shown.

APPENDIX

B

THE TECHNICOLOR TG788VN V2 MEDIAACCESS GATEWAY

The CPE used in this project is the Technicolor TG788vn v2 MediaAccess Gateway (MAG) that is a xDSL modem provided by Telenor to their residential customers in order to support high-speed network access. As other residential or Small Office Home Office (SOHO) routers, the MAG has multiple devices integrated that have functionalities like router, switch, xDSL modem, and wireless access point. The main features of the MAG are as follows:

Broadband access: DSL port allows to connect the gateway to the ISP DSL connection. It has an integrated modem supporting multi xDSL standards.

Wired access: Built-in Ethernet switch with four fast Ethernet ports that allows to connect multiple devices to the LAN network.

Wireless access: IEEE 802.11b/g/n (2.4 GHz) wireless access point that supports multiple SSIDs with possibility of creating up to four WLANs.

USB port: Allows to connect a mobile USB adapter to achieve an alternative 3G Internet access as well as connecting printers or file servers for media sharing.

VoIP: Has enabled VoIP services for IP phones. It contains two phone POTS ports for regular phones and faxes allowing them to communicate via the Internet.

Security: Parental control, Content-based and address-based filtering, Firewall, NAT, and services like Application Level Gateways (ALGs), Packet Inspection (SPI) and Intrusion Detection and Prevention System (IDS). The gateway also supports WPA2, WPS protocols and Demilitarized Zone (DMZ).

QoS: Ensures different levels of QoS including IP QoS, Ethernet QoS and Wireless QoS.

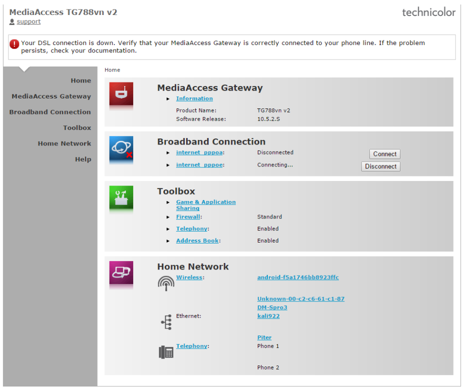
802.1q support: Supports VLANs, VLAN bridging and multiple bridge instances.

IPv6 support: Gold IPv6 Ready certification ensuring that the device deployed is capable of IPv6.

The MAG can be accessed, configured or managed by the xDSL subscriber in two ways. The methods are described below and can be seen in Figures B.1 and B.2:

MediaAccess Gateway GUI: A graphical interface can be accessed through a web browser in order to configure the modem in an easy and clear manner. It is the appropriate way for xDSL subscribers with limited knowledge about CPE configurations.

Command Line Interface (CLI): A telnet session interface can be accessed through a telnet client, and allows to configure the MAG in a more expertly manner using CLI commands. Using CLI command gives more configuring possibilities and options. so that it is adequate for a more complicated and in-depth networking configuration. *The CLI command interface will/has been used in the project.*



MediaAccess TG788vn v2

support

Your DSL connection is down. Verify that your MediaAccess Gateway is correctly connected to your phone line. If the problem persists, check your documentation.

Home

MediaAccess Gateway

- Information

Product Name: TG788vn v2
Software Release: 10.5.2.S

Broadband Connection

- internet_pppoa: Disconnected
- internet_pppoe: Connecting...

Toolbox

- Game & Application Sharing
- Firewall: Standard
- Telephony: Enabled
- Address Book: Enabled

Home Network

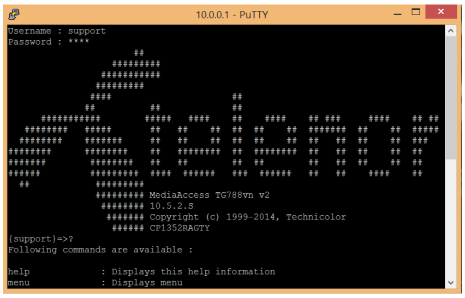
Wireless: android-f5a1746bb8923ffc

Ethernet: Unknown-00-c2-c6-61-c1-87
DM-Spro3
kali922

Telephony: Piter
Phone 1

Phone 2

Figure B.1: Graphical User Interface



```
Username : support
Password : ****
                       ##
                   #########
                 ###########
                #########
               ####                      ##
              ##          ##             ##
      ###########        #####   ####    ##    ####    ## ###     ####    ## ##
   ########   #####       ##   ##    ##  ##  ##    ##  #######  ##    ##  #####
  ########    #######     ##   ##    ##  ##  ##    ##  ##   ##  ##    ##  ###
########      ########    ##   ########  ##  ########  ##   ##  ##    ##  ##
#######        ########  ##    ##        ##  ##        ##   ##  ##    ##  ##
######         #########  ####  ######   ###  ######   ##   ##    ####    ##
  ##            #########
                ######### MediaAccess TG788vn v2
                 ######## 10.5.2.S
                  ####### Copyright (c) 1999-2014, Technicolor
                   ###### CP1352RAGTY
{support}=>?
Following commands are available :

help             : Displays this help information
menu             : Displays menu
```

Figure B.2: Command Line Interface

APPENDIX

C

LISTS OF MAG SETUP

This appendix includes the lists obtained by the CLI command:
=> *interface list expand=disabled reverse=enabled*

1. List of default interface connections
2. List of interface connections after the setup in Chapter 5

The lists along with figures (based on the lists) can be seen in the following pages.

Default interface connections in the MAG

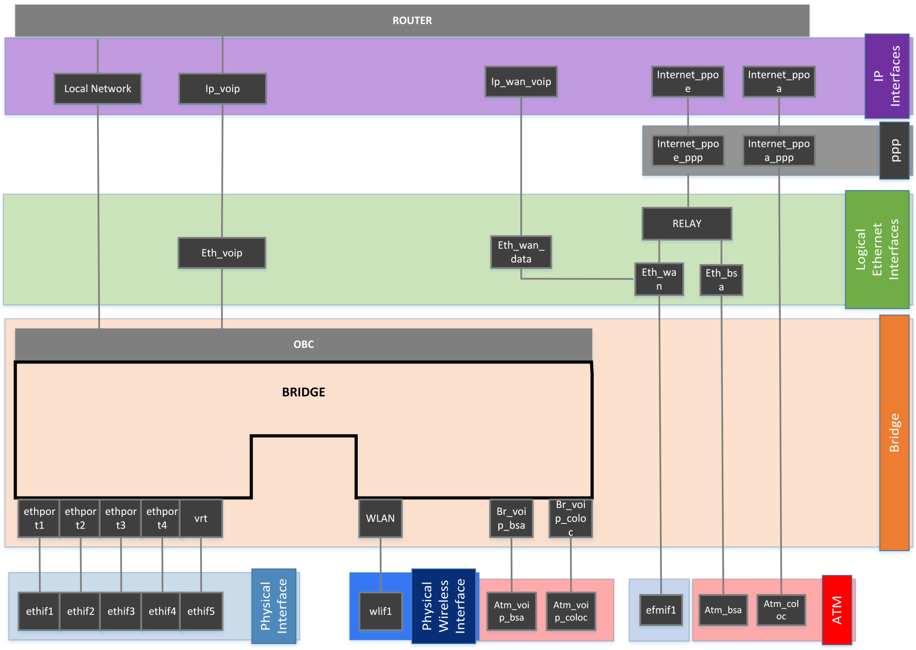


Figure C.1: Logical Ethernet and Interface Architecture

```
=> interface list expand=disabled reverse=enabled
Name                Type      State          Use   LL Interfaces
----                ----      -----          ---   -------------
loop                ip        connected      0
ethif1              physical  connected      1    ethport1
ethif2              physical  connected      1    ethport2
ethif3              physical  connected      1    ethport3
ethif4              physical  connected      1    ethport4
efmif1              physical  connected      1    eth_wan
ethif5              physical  connected      1    virt
ndisveth0           physical  connected      0
bridge              eth       connected      3    eth_voip,
                                                  LocalNetwork
OBC                 bridge    connected      1    bridge
ethport1            bridge    connected      1    bridge
RELAY               eth       connected      1    internet_pppoe_ppp
atm_coloc           atm       connected      1    internet_pppoa_ppp
atm_bsa             atm       connected      1    eth_bsa
atm_voip_bsa        atm       connected      1    br_voip_bsa
atm_voip_coloc      atm       connected      1    br_voip_coloc
wlif1               physical  connected      1    WLAN
ethport2            bridge    connected      1    bridge
ethport3            bridge    connected      1    bridge
ethport4            bridge    connected      1    bridge
WLAN                bridge    connected      1    bridge
virt                bridge    connected      1    bridge
br_voip_bsa         bridge    connected      1    bridge
```

```
br_voip_coloc        bridge    connected       1   bridge
eth_wan              eth       connected       2   eth_wan_data,
                                                   eth_wan_voip
eth_wan_data         eth       connected       1   RELAY
eth_wan_voip         eth       connected       1   ip_wan_voip
eth_bsa              eth       connected       1   RELAY
eth_voip             eth       connected       1   ip_voip
internet_pppoa_ppp   ppp       connected       1   internet_pppoa
internet_pppoa       ip        not-connected   0
internet_pppoe_ppp   ppp       connecting      1   internet_pppoe
internet_pppoe      *ip        not-connected   0
ip_voip              ip        not-connected   0
LocalNetwork         ip        connected       0
ip_wan_voip          ip        not-connected   0
FXS1                 physical  connected       0
FXS2                 physical  connected       0
```

Listing C.1: Default interfaces list on Technicolor MediaAccess Gateway TG788vn v2.

Interface connections after the setup of Part I

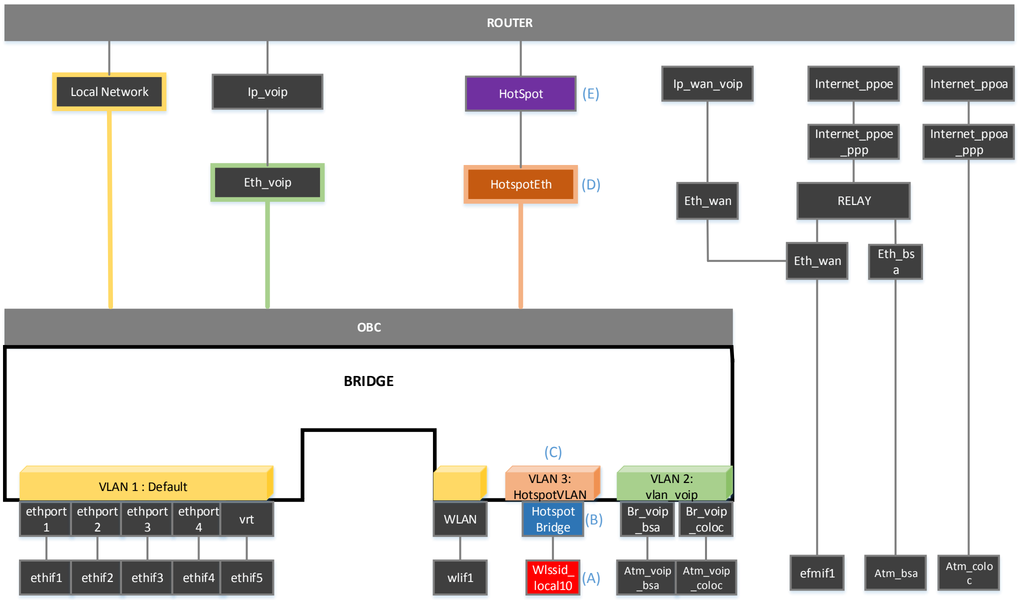


Figure C.2: Solution design of the MAG.

```
=> interface list expand=disabled reverse=enabled
Name                Type      State          Use  LL Interfaces
----                ----      -----          ---  -------------
loop                ip        connected      0
ethif1              physical  connected      1   ethport1
ethif2              physical  connected      1   ethport2
ethif3              physical  connected      1   ethport3
ethif4              physical  connected      1   ethport4
efmif1              physical  connected      1   eth_wan
ethif5              physical  connected      1   virt
ndisveth0           physical  connected      0
bridge              eth       connected      3   eth_voip,
                                                 LocalNetwork,
                                                 HotspotEth
OBC                 bridge    connected      1   bridge
ethport1            bridge    connected      1   bridge
RELAY               eth       connected      1   internet_pppoe_ppp
atm_coloc           atm       connected      1   internet_pppoa_ppp
atm_bsa             atm       connected      1   eth_bsa
atm_voip_bsa        atm       connected      1   br_voip_bsa
atm_voip_coloc      atm       connected      1   br_voip_coloc
wlif1               physical  connected      1   WLAN
ethport2            bridge    connected      1   bridge
ethport3            bridge    connected      1   bridge
ethport4            bridge    connected      1   bridge
WLAN                bridge    connected      1   bridge
virt                bridge    connected      1   bridge
br_voip_bsa         bridge    connected      1   bridge
br_voip_coloc       bridge    connected      1   bridge
eth_wan             eth       connected      2   eth_wan_data,
                                                 eth_wan_voip
eth_wan_data        eth       connected      1   RELAY
eth_wan_voip        eth       connected      1   ip_wan_voip
eth_bsa             eth       connected      1   RELAY
eth_voip            eth       connected      1   ip_voip
internet_pppoa_ppp  ppp       connected      1   internet_pppoa
internet_pppoa      ip        not-connected  0
internet_pppoe_ppp  ppp       connecting     1   internet_pppoe
internet_pppoe     *ip        not-connected  0
ip_voip             ip        not-connected  0
LocalNetwork        ip        connected      0
ip_wan_voip         ip        not-connected  0
FXS1                physical  connected      0
FXS2                physical  connected      0
wl_ssid1_local0     physical  connected      1   HotspotBridge
HotspotBridge       bridge    connected      1   bridge
HotspotEth          eth       connected      1   HotSpotIP
HotSpotIP           ip        connected      0
```

Listing C.2: Interfaces list on Technicolor MediaAccess Gateway TG788vn v2 when the configuration has been done.

APPENDIX

D

FIREWALL RULES

This Appendix includes:

Section D.1 on the next page	**: Firewall rules before configuration.**
Section D.2 on page 212	**: Firewall rules after configuration.**
Section D.3 on page 215	**: List of expressions**

The firewall has three different kinds of packet flows:

forward: Traffic from an interface to another interface.

source: Traffic from the MAG

sink: Traffic to the MAG

When a packet is received it will be classified as one of these flows and go into one of the *chains*: *sink*, *forward* or *source*. In that flow the firewall will determine from the conditions, which chains the packet will be forwarded to (*link*). *forward* can e.g. *link* to *forward_level*, which includes:

forward_level_BlockAll: Block all traffic to/from the Internet.

forward_level_Standard: Allow outgoing connections and block incoming traffic.

forward_level_Disable: Disable the firewall. Allow all traffic.

The firewall is read top-down, which means the index with lowest number has highest priority when conditions are conflicting.

D.1 Firewall rules before configuration

```
=> firewall rule list
Rules (flags: C=Constant, D=Dynamic, E=Enable, L=Log)
=====
Chain  Nr.  Flags  Rule name         Action                       Conditions
-----  ---  -----  ---------         ------                       ----------
sink
       1    CDE                    : link sink_fire               : *.* > *.*
       2    CDE                    : link sink_system_service     : *.* > *.*
       3    CDEL   ResourceGate    : accept                       : ResourceGate_sv *.* > *.*
forward
       1    CDE                    : link forward_multicast       : *.* > *.*
       2    CDE    For_NAT_port... : link forward_portmapping     : *.* > *.*
       3    CDE                    : link forward_multicast6      : *.* > *.*
       4    CDE                    : link forward_fire            : *.* > *.*
       5    CDE                    : link forward_timeofday       : *.* > *.*
       6    CDE                    : link forward_host_service    : *.* > *.*
       7    CDE                    : link forward_custom          : *.* > *.*
       8    CDE                    : link forward_level           : *.* > *.*
source
       1    CDE                    : link source_fire             : *.* > *.*
       2    CDE                    : link source_system_service   : *.* > *.*
source_fire
       1    C E    AnyTraffic      : accept                       : *.* > *.*
sink_system_service
       1    C E    https_wan_2033  : accept                       : wan_https_2033 *.* > *.*
       2    C E    https_wan_2034  : accept                       : wan_https_2034 *.* > *.*
       3    C E    http_wan_2033   : accept                       : wan_http_2033 *.* > *.*
       4    C E    http_wan_2034   : accept                       : wan_http_2034 *.* > *.*
       5    CDE                    : accept                       : DHCP-C_sv_in *._ipv4_addr > *.*
       6    CDE                    : accept                       : DHCPv6-R-C_sv_in DHCPv6-R-C_if_in._ipv6_addr >
```

```
                                                                      *.*
       7    CDE                     : accept                        : DHCPv6-R-SR_sv_in DHCPv6-R-SR_if_in._ipv6_addr >
                                                                      *.*
       8    CDE                     : accept                        : UPnP-HTTP_sv_in UPnP-HTTP_if_in._ipv4_addr > *.*
       9    CDE                     : accept                        : CWMP-S_sv_in CWMP-S_if_in._ipv4_addr > *.*
       10   CDE                     : accept                        : SSDP_sv_in SSDP_if_in._ipv4_addr > *.*
       11   CDE                     : accept                        : PPTPD_sv_in PPTPD_if_in._ipv4_addr > *.*
       12   CDE                     : accept                        : PPTPGRE_sv PPTPGRE_if_in._ipv4_addr > *.*
       13   CDE                     : accept                        : HTTPs_sv_in HTTPs_if_in.* > *.*
       14   CDE                     : accept                        : FTP_sv_in FTP_if_in.* > *.*
       15   CDE                     : accept                        : TELNET_sv_in TELNET_if_in.* > *.*
       16   CDE                     : accept                        : IGMP-Proxy_sv *._ipv4_addr > *.*
       17   CDE                     : accept                        : DHCPv6-S_sv_in DHCPv6-S_if_in._ipv6_addr > *.*
       18   CDE                     : accept                        : MDAP_sv_in MDAP_if_in._ipv4_addr > *.*
       19   CDE                     : accept                        : VOIP_SIP_sv *._ipv4_addr > *.*
       20   CDE                     : accept                        : VOIP_SIP_UDP_sv_in *._ipv4_addr > *.*
       21   CDE                     : accept                        : VOIP_SIP_SIG_sv *._ipv4_addr > *.*
       22   CDE                     : accept                        : PING_RESPONDER_sv PING_RESPONDER_if_in._ipv4_addr
                                                                      > *.*
       23   CDE                     : accept                        : PINGv6_RESP_sv PINGv6_RESP_if_in._ipv6_addr > *.*
       24   CDE                     : accept                        : NDSOL_sv *._ipv6_addr > *.*
       25   CDE                     : accept                        : NDADV_sv *._ipv6_addr > *.*
       26   CDE                     : accept                        : RTSOL_sv RTSOL_if_in._ipv6_addr > *.*
       27   CDE                     : accept                        : RTADV_sv *.RTADV_ip_in > *.*
       28   CDE                     : accept                        : DNS-S-UDP_sv_in DNS-S-UDP_if_in.* > *.*
       29   CDE                     : accept                        : DNS-S-TCP_sv_in DNS-S-TCP_if_in.* > *.*
       30   CDE                     : accept                        : DNS-S_sv DNS-S_if_in.* > *.*
       31   CDE                     : accept                        : DHCP-S_sv_in DHCP-S_if_in._ipv4_addr > *.*
       32   CDE                     : accept                        : DHCP-R_sv_in DHCP-R_if_in._ipv4_addr > *.*
forward_level
       1    CDE                     : link forward_level_Standard   : *.* > *.*
       2    C E    ContentFTP       : accept                        : ftp wan.* > lan.contentsharing_ip
forward_level_BlockAll
```

```
       1    C E    AnyTraffic       : drop                         : *.* > *.*
forward_level_Standard
       1    C E    FromLAN          : accept                       : lan.* > *.*
forward_level_Disabled
       1    C E    AnyTraffic       : accept                       : *.* > *.*
forward_portmapping
       1    CDE    FORW_PM          : accept                       : SERV_INP_17_62014_62014 SRCIF_IP-Intf-
                                                                     internet_pppoe.* > *.DSTIP_10-0-0-11
       2    CDE    FORW_PM          : accept                       : SERV_INP_17_6984_6984 SRCIF_IP-Intf-
                                                                     internet_pppoe.* > *.DSTIP_10-0-0-11
       3    CDE    FORW_PM          : accept                       : SERV_INP_6_6984_6984 SRCIF_IP-Intf-
                                                                     internet_pppoe.* > *.DSTIP_10-0-0-11
```

Listing D.1: All firewall rules implemented in the MAG

D.2 Firewall rules after configuration

```
=> firewall rule list
Rules (flags: C=Constant, D=Dynamic, E=Enable, L=Log)
=====
Chain  Nr.  Flags  Rule name         Action                       Conditions
-----  ---  -----  ---------         ------                       ----------
sink
       1    CDE                    : link sink_fire               : *.* > *.*
       2    CDE                    : link sink_system_service     : *.* > *.*
       3    CDEL   ResourceGate    : accept                       : ResourceGate_sv *.* > *.*
forward
       1    CDE                    : link forward_multicast       : *.* > *.*
       2    CDE    For_NAT_port... : link forward_portmapping     : *.* > *.*
       3    CDE                   : link forward_multicast6      : *.* > *.*
       4    CDE                   : link forward_fire            : *.* > *.*
       5    CDE                   : link forward_timeofday       : *.* > *.*
       6    CDE                   : link forward_host_service    : *.* > *.*
       7    CDE                   : link forward_custom          : *.* > *.*
       8    CDE                   : link forward_level           : *.* > *.*
source
       1    CDE                   : link source_fire             : *.* > *.*
       2    CDE                   : link source_system_service   : *.* > *.*
source_fire
       1    C E    AnyTraffic     : accept                       : *.* > *.*
sink_system_service
       1    C E    HSgateway      : deny                         : lan.* > *.192.168.13.33

       2    C E    dmzicmp        : accept                       : icmp dmz.* > *.192.168.13.33
       3    C E    dmzdhcp        : accept                       : dhcp dmz.* > *.*

       4    C E    dmzdns         : accept                       : dns dmz.* > *.192.168.13.33
       5    C E    dmzall         : deny                         : dmz.* > *.*
       6    C E    https_wan_2033 : accept                       : wan_https_2033 *.* > *.*
       7    C E    https_wan_2034 : accept                       : wan_https_2034 *.* > *.*
       8    C E    http_wan_2033  : accept                       : wan_http_2033 *.* > *.*
       9    C E    http_wan_2034  : accept                       : wan_http_2034 *.* > *.*
       10    CDE                   : accept                       : DHCP-C_sv_in *._ipv4_addr > *.*
       11    CDE                   : accept                       : DHCPv6-R-C_sv_in DHCPv6-R-C_if_in._ipv6_addr >
                                                                  *.*
       12   CDE                   : accept                       : DHCPv6-R-SR_sv_in DHCPv6-R-SR_if_in._ipv6_addr >
                                                                   *.*
       13   CDE                   : accept                       : UPnP-HTTP_sv_in UPnP-HTTP_if_in._ipv4_addr > *.*
       14   CDE                   : accept                       : CWMP-S_sv_in CWMP-S_if_in._ipv4_addr > *.*
       15   CDE                   : accept                       : SSDP_sv_in SSDP_if_in._ipv4_addr > *.*
       16   CDE                   : accept                       : PPTPD_sv_in PPTPD_if_in._ipv4_addr > *.*
       17   CDE                   : accept                       : PPTPGRE_sv PPTPGRE_if_in._ipv4_addr > *.*
       18   CDE                     : accept                       : HTTPs_sv_in HTTPs_if_in.* > *.*
       19   CDE                     : accept                       : FTP_sv_in FTP_if_in.* > *.*
       20   CDE                     : accept                       : TELNET_sv_in TELNET_if_in.* > *.*
       21   CDE                     : accept                       : IGMP-Proxy_sv *._ipv4_addr > *.*
       22   CDE                     : accept                       : DHCPv6-S_sv_in DHCPv6-S_if_in._ipv6_addr > *.*
       23   CDE                     : accept                       : MDAP_sv_in MDAP_if_in._ipv4_addr > *.*
       24   CDE                     : accept                       : VOIP_SIP_sv *._ipv4_addr > *.*
       25   CDE                     : accept                       : VOIP_SIP_UDP_sv_in *._ipv4_addr > *.*
       26   CDE                     : accept                       : VOIP_SIP_SIG_sv *._ipv4_addr > *.*
       27   CDE                     : accept                       : PING_RESPONDER_sv PING_RESPONDER_if_in._ipv4_addr
                                                                     > *.*
       28   CDE                     : accept                       : PINGv6_RESP_sv PINGv6_RESP_if_in._ipv6_addr > *.*
       29   CDE                     : accept                       : NDSOL_sv *._ipv6_addr > *.*
       30   CDE                     : accept                       : NDADV_sv *._ipv6_addr > *.*
       31   CDE                     : accept                       : RTSOL_sv RTSOL_if_in._ipv6_addr > *.*
       32   CDE                     : accept                       : RTADV_sv *.RTADV_ip_in > *.*
       33   CDE                     : accept                       : DNS-S-UDP_sv_in DNS-S-UDP_if_in.* > *.*
       34   CDE                     : accept                       : DNS-S-TCP_sv_in DNS-S-TCP_if_in.* > *.*
       35   CDE                     : accept                       : DNS-S_sv DNS-S_if_in.* > *.*
       36   CDE                     : accept                       : DHCP-S_sv_in DHCP-S_if_in._ipv4_addr > *.*
       37   CDE                     : accept                       : DHCP-R_sv_in DHCP-R_if_in._ipv4_addr > *.*
forward_level
       1    CDE                     : link forward_level_Standard  : *.* > *.*
       2    C E    ContentFTP       : accept                       : ftp wan.* > lan.contentsharing_ip
forward_level_BlockAll
       1    C E    AnyTraffic       : drop                         : *.* > *.*
forward_level_Standard
       1    C E    HStoWAN          : accept                       : dmz.* > wan.*
       2    C E    ALLtoHS          : deny                         : *.* > dmz.*
       3    C E    HStoALL          : deny                         : dmz.* > *.*
       4    C E    FromLAN          : accept                       : lan.* > *.*
forward_level_Disabled
       1    C E    AnyTraffic       : accept                       : *.* > *.*
```

```
forward_portmapping
       1    CDE    FORW_PM         : accept                         : SERV_INP_17_62014_62014 SRCIF_IP-Intf-
                                                                      internet_pppoe.* > *.DSTIP_10-0-0-11
       2    CDE    FORW_PM         : accept                         : SERV_INP_17_6984_6984 SRCIF_IP-Intf-
                                                                      internet_pppoe.* > *.DSTIP_10-0-0-11
       3    CDE    FORW_PM         : accept                         : SERV_INP_6_6984_6984 SRCIF_IP-Intf-
                                                                      internet_pppoe.* > *.DSTIP_10-0-0-11
```

Listing D.2: A list of the default firewall rules in MAG

D.3 List of Expressions

proto meaning which protocol is used [IANA [2015]]:
proto=1: ICMP
proto=6: TCP
proto=17: UDP

```
=>expr list
name                                     type    use flags    expression
----                                     ----    ---------    ----------
wan                                      intf       5         1. intfgroup=0
local                                    intf       3         1. intfgroup=1
lan                                      intf       2         1. intfgroup=2
tunnel                                   intf       0         1. intfgroup=3
dmz                                      intf       7         1. intfgroup=4
guest                                    intf       0         1. intfgroup=5
contentsharing_ip                        ip         1         1. addr=10.0.0.254
```

```
private                                  ip         0        1. addr=10.0.0.0/8
                                                             2. addr=172.[16-31].*.*
                                                             3. addr=192.168.1.0/24
ssdp_ip                                  ip         0        1. addr=239.255.255.250
mdap_ip                                  ip         0        1. addr=224.0.0.103
192.168.13.[37-46]                       ip         2        1. addr=192.168.13.[37-46]
10.0.0.[2-254]                           ip         2        1. addr=10.0.0.[2-254]
192.168.13.33                            ip         2        1. addr=192.168.13.33
wan_https_2033                           serv       1        1. proto=6 dst-prt=2033
wan_https_2034                           serv       1        1. proto=6 dst-prt=2034
wan_http_2033                            serv       1        1. proto=6 dst-prt=2033
wan_http_2034                            serv       1        1. proto=6 dst-prt=2034
icmp                                     serv       2        1. proto=1
igmp                                     serv       1        1. proto=2
ftp                                      serv       1        1. proto=6 dst-prt=21
telnet                                   serv       1        1. proto=6 dst-prt=23
http                                     serv       1        1. proto=6 dst-prt=80
httpproxy                                serv       1        1. proto=6 dst-prt=8080
https                                    serv       1        1. proto=6 dst-prt=443
RPC                                      serv       0        1. proto=6 dst-prt=135
NBT                                      serv       0        1. proto=17 dst-prt=137
                                                             2. proto=17 dst-prt=138
                                                             3. proto=6 dst-prt=139
SMB                                      serv       0        1. proto=6 dst-prt=445
imap                                     serv       1        1. proto=6 dst-prt=143
imap3                                    serv       1        1. proto=6 dst-prt=220
imap4-ssl                                serv       1        1. proto=6 dst-prt=585
imaps                                    serv       1        1. proto=6 dst-prt=993
pop2                                     serv       1        1. proto=6 dst-prt=109
pop3                                     serv       1        1. proto=6 dst-prt=110
pop3s                                    serv       1        1. proto=6 dst-prt=995
smtp                                     serv       1        1. proto=6 dst-prt=25
ssh                                      serv       0        1. proto=6 dst-prt=22
```

```
dns                                     serv       2        1. proto=6 dst-prt=53
                                                            2. proto=17 dst-prt=53
nntp                                    serv       0        1. proto=6 dst-prt=119
ipsec                                   serv       0        1. proto=51
                                                            2. proto=50
                                                            3. proto=17 dst-prt=500
                                                            4. proto=17 dst-prt=4500
esp                                     serv       1        1. proto=50
ah                                      serv       1        1. proto=51
ike                                     serv       1        1. proto=17 dst-prt=500
sip                                     serv       1        1. proto=17 dst-prt=5060
                                                            2. proto=6 dst-prt=5060
h323                                    serv       0        1. proto=6 dst-prt=1720
                                                            2. proto=17 dst-prt=1720
                                                            3. proto=6 dst-prt=1718
                                                            4. proto=17 dst-prt=1718
                                                            5. proto=6 dst-prt=1719
                                                            6. proto=17 dst-prt=1719
dhcp                                    serv       2        1. proto=17 dst-prt=68
                                                            2. proto=17 dst-prt=67
rtsp                                    serv       1        1. proto=17 dst-prt=554
                                                            2. proto=6 dst-prt=554
ssdp_serv                               serv       0        1. proto=17 dst-prt=1900
mdap_serv                               serv       0        1. proto=17 dst-prt=3235
syslog                                  serv       0        1. proto=17 dst-prt=514
VoIP-Inc-SIP-UDP                        serv       1        1. proto=17 dst-prt=5060
VoIP-Inc-SIP-TCP                        serv       1        1. proto=6 dst-prt=5060
VoIP-Inc-RTP                            serv       1        1. proto=17 dst-prt=1024->1151
gre                                     serv       0        1. proto=47
icmpv6                                  serv       1        1. proto=58
```

Listing D.3: A list of expressions defined in the CPE

APPENDIX E

NMAP SCANS

Nmap scans are done for the following networks:

1. Nmap scan from the private network
2. Nmap scan from the hotspot network

This Appendix includes the following Sections:

- ICMP echo (ping) scan

Private: Appendix E.0.1 on the following page
Hotspot: Appendix E.0.3 on page 224

- TCP port scan

Private$^\diamond$: Appendix E.0.2 on the following page
Hotspot$^\diamond$: Appendix E.0.4 on page 224

- UDP port scan

Private$^\diamond$: Appendix E.0.2 on the following page
Hotspot$^\diamond$: Appendix E.0.4 on page 224

$^\diamond$: TCP and UDP scans are done at once.

Nmap scan from the private network

E.0.1 Private Ping Scan

```
root@kali922:~# nmap -sn 192.168.13.33 192.168.13.37-46 10.0.0.1
    10.0.0.2-20 85.81.240.13

Starting Nmap 6.49BETA5 ( https://nmap.org ) at 2015-12-07 10:40
    CET
Nmap scan report for dsldevice.lan (192.168.13.33)
Host is up (0.053s latency).

Nmap scan report for dsldevice.lan (10.0.0.1)
Host is up (0.0017s latency).
MAC Address: 30:91:8F:94:46:7A (Technicolor)

Nmap scan report for RPi2.lan (10.0.0.2)
Host is up (0.0030s latency).
MAC Address: E8:94:F6:1C:1A:06 (Tp-link Technologies Co.)

Nmap scan report for Win1.lan (10.0.0.3)
Host is up (0.032s latency).
MAC Address: E8:94:F6:1B:F6:6A (Tp-link Technologies Co.)

Nmap scan report for 10.0.0.4
Host is up (0.028s latency).
MAC Address: E8:94:F6:1B:F6:6A (Tp-link Technologies Co.)

Nmap scan report for 10.0.0.5
Host is up (0.0046s latency).
MAC Address: B8:27:EB:9C:C9:05 (Raspberry Pi Foundation)

Nmap scan report for kali922.lan (10.0.0.7)
Host is up.

Nmap scan report for dsldevice.lan (85.81.240.13)
Host is up (0.0017s latency).

Nmap done: 32 IP addresses (7 hosts up) scanned in 4.27 seconds
```

Listing E.1: Nmap ping scan from the private network

E.0.2 Private TCP and UDP Scan

```
root@kali922:~# nmap -Pn -sS -sU -p 1- 10.0.0.1 10.0.0.2-15
    85.81.240.13 192.168.13.33 192.168.13.37-46

Starting Nmap 6.49BETA5 ( https://nmap.org ) at 2015-12-17 14:43
    CET
Nmap scan report for dsldevice.lan (10.0.0.1)
Host is up (0.0041s latency).
Not shown: 65534 open|filtered ports, 65528 filtered ports
```

```
PORT      STATE  SERVICE
53/tcp    open   domain
443/tcp   open   https
1723/tcp  open   pptp
2034/tcp  open   scoremgr
2121/tcp  open   ccproxy-ftp
9001/tcp  closed tor-orport
23023/tcp open   unknown
53/udp    open   domain
MAC Address: 30:91:8F:94:46:7A (Technicolor)

Nmap scan report for RPi2.lan (10.0.0.2)
Host is up (0.058s latency).
Not shown: 131064 closed ports
PORT      STATE         SERVICE
22/tcp    open          ssh
68/udp    open|filtered dhcpc
123/udp   open          ntp
546/udp   open|filtered dhcpv6-client
5353/udp  open          zeroconf
57753/udp open|filtered unknown
MAC Address: E8:94:F6:1C:1A:06 (Tp-link Technologies Co.)

Nmap scan report for Win1.lan (10.0.0.3)
Host is up (0.013s latency).
Not shown: 65534 open|filtered ports, 65528 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
554/tcp   open  rtsp
2869/tcp  open  icslap
5357/tcp  open  wsdapi
10243/tcp open  unknown
137/udp   open  netbios-ns
MAC Address: E8:94:F6:1C:1A:06 (Tp-link Technologies Co.)

Nmap scan report for RPi3.lan (10.0.0.4)
Host is up (0.0082s latency).
Not shown: 131064 closed ports
PORT      STATE         SERVICE
22/tcp    open          ssh
68/udp    open|filtered dhcpc
123/udp   open          ntp
546/udp   open|filtered dhcpv6-client
5353/udp  open          zeroconf
48105/udp open|filtered unknown
MAC Address: E8:94:F6:1B:F6:6A (Tp-link Technologies Co.)

Nmap scan report for 10.0.0.5
Host is up (0.0053s latency).
Not shown: 131064 closed ports
PORT      STATE         SERVICE
22/tcp    open          ssh
68/udp    open|filtered dhcpc
123/udp   open          ntp
546/udp   open|filtered dhcpv6-client
```

```
5353/udp  open          zeroconf
39422/udp open|filtered unknown
MAC Address: B8:27:EB:9C:C9:05 (Raspberry Pi Foundation)

Nmap scan report for kali922.lan (10.0.0.7)
Host is up (0.000013s latency).
Not shown: 131065 closed ports
PORT      STATE         SERVICE
68/udp    open|filtered dhcpc
69/udp    open|filtered tftp
5353/udp  open|filtered zeroconf
49943/udp open|filtered unknown
50319/udp open|filtered unknown

Warning: 192.168.13.38 giving up on port because retransmission cap
     hit (10).
Warning: 192.168.13.39 giving up on port because retransmission cap
     hit (10).
Warning: 192.168.13.37 giving up on port because retransmission cap
     hit (10).
Warning: 192.168.13.42 giving up on port because retransmission cap
     hit (10).
Warning: 192.168.13.45 giving up on port because retransmission cap
     hit (10).
Warning: 192.168.13.41 giving up on port because retransmission cap
     hit (10).
Warning: 192.168.13.46 giving up on port because retransmission cap
     hit (10).
Warning: 192.168.13.33 giving up on port because retransmission cap
     hit (10).
Warning: 192.168.13.43 giving up on port because retransmission cap
     hit (10).
Warning: 192.168.13.44 giving up on port because retransmission cap
     hit (10).
Warning: 192.168.13.40 giving up on port because retransmission cap
     hit (10).
Nmap scan report for dsldevice.lan (85.81.240.13)
Host is up (0.030s latency).
Not shown: 65534 open|filtered ports, 65526 filtered ports
PORT      STATE  SERVICE
53/tcp    open   domain
443/tcp   open   https
1723/tcp  open   pptp
2033/tcp  open   glogger
2034/tcp  open   scoremgr
2121/tcp  open   ccproxy-ftp
8000/tcp  open   http-alt
9001/tcp  closed tor-orport
23023/tcp open   unknown
53/udp    open   domain

Nmap scan report for 192.168.13.33
Host is up (0.0021s latency).
All 131070 scanned ports on 192.168.13.33 are filtered (130449) or
    open|filtered (621)

Nmap scan report for 192.168.13.37
```

```
Host is up (0.0048s latency).
All 131070 scanned ports on 192.168.13.37 are filtered (130374) or
    open|filtered (696)

Nmap scan report for 192.168.13.38
Host is up (0.0012s latency).
All 131070 scanned ports on 192.168.13.38 are filtered (130511) or
    open|filtered (559)

Nmap scan report for 192.168.13.39
Host is up (0.0012s latency).
All 131070 scanned ports on 192.168.13.39 are filtered (130432) or
    open|filtered (638)

Nmap scan report for 192.168.13.40
Host is up (0.0013s latency).
All 131070 scanned ports on 192.168.13.40 are filtered (130484) or
    open|filtered (586)

Nmap scan report for 192.168.13.41
Host is up (0.0011s latency).
All 131070 scanned ports on 192.168.13.41 are filtered (130468) or
    open|filtered (602)

Nmap scan report for 192.168.13.42
Host is up (0.0012s latency).
All 131070 scanned ports on 192.168.13.42 are filtered (130434) or
    open|filtered (636)

Nmap scan report for 192.168.13.43
Host is up (0.022s latency).
All 131070 scanned ports on 192.168.13.43 are filtered (130406) or
    open|filtered (664)

Nmap scan report for 192.168.13.44
Host is up (0.0020s latency).
All 131070 scanned ports on 192.168.13.44 are filtered (130438) or
    open|filtered (632)

Nmap scan report for 192.168.13.45
Host is up (0.0025s latency).
All 131070 scanned ports on 192.168.13.45 are filtered (130352) or
    open|filtered (718)

Nmap scan report for 192.168.13.46
Host is up (0.0025s latency).
All 131070 scanned ports on 192.168.13.46 are filtered (130407) or
    open|filtered (663)

Nmap done: 27 IP addresses (18 hosts up) scanned in 292265.38
    seconds
```

Listing E.2: Nmap TCP and UDP scan (done in one) from the private network

Nmap scan from the hotspot network

E.0.3 Hotspot Ping Scan

```
root@kali922:~# nmap -sn 192.168.13.33 192.168.13.37-46 10.0.0.1
    10.0.0.2-20 85.81.240.13

Starting Nmap 6.49BETA5 ( https://nmap.org ) at 2015-12-04 16:12
    CET
Nmap scan report for dsldevice.lan (192.168.13.33)
Host is up (0.00088s latency).
MAC Address: 30:91:8F:94:46:7A (Technicolor)

Nmap scan report for kali922.lan (192.168.13.41)
Host is up.

Note: Host seems down. If it is really up, but blocking our ping
    probes, try -Pn
Nmap done: 32 IP address (2 hosts up) scanned in 10.99 seconds
```

Listing E.3: Nmap ping scan from the hotspot network

E.0.4 Hotspot TCP and UDP Scan

```
root@kali922:~# nmap -Pn -sS -sU -p 1- 10.0.0.1 10.0.0.2-15
    85.81.240.13 192.168.13.33 192.168.13.37-46

Starting Nmap 6.49BETA5 ( https://nmap.org ) at 2015-12-27 12:02
    CET
Warning: 10.0.0.5 giving up on port because retransmission cap hit
    (10).
Warning: 10.0.0.3 giving up on port because retransmission cap hit
    (10).
Warning: 10.0.0.2 giving up on port because retransmission cap hit
    (10).
Warning: 10.0.0.6 giving up on port because retransmission cap hit
    (10).
Warning: 10.0.0.12 giving up on port because retransmission cap hit
     (10).
Warning: 85.81.240.13 giving up on port because retransmission cap
    hit (10).
Warning: 10.0.0.9 giving up on port because retransmission cap hit
    (10).
Warning: 10.0.0.4 giving up on port because retransmission cap hit
    (10).
Warning: 10.0.0.13 giving up on port because retransmission cap hit
     (10).
Warning: 10.0.0.7 giving up on port because retransmission cap hit
    (10).
Warning: 10.0.0.1 giving up on port because retransmission cap hit
    (10).
```

```
Warning: 10.0.0.15 giving up on port because retransmission cap hit
     (10).
Warning: 10.0.0.14 giving up on port because retransmission cap hit
     (10).
Warning: 10.0.0.11 giving up on port because retransmission cap hit
     (10).
Warning: 10.0.0.10 giving up on port because retransmission cap hit
     (10).
Warning: 10.0.0.8 giving up on port because retransmission cap hit
    (10).
Nmap scan report for dsldevice.lan (10.0.0.1)
Host is up (0.0024s latency).
All 131070 scanned ports on dsldevice.lan (10.0.0.1) are filtered
    (127659) or open|filtered (3411)

Nmap scan report for RPi2.lan (10.0.0.2)
Host is up (0.0036s latency).
All 131070 scanned ports on RPi2.lan (10.0.0.2) are filtered
    (127554) or open|filtered (3516)

Nmap scan report for Win1.lan (10.0.0.3)
Host is up (0.0049s latency).
Not shown: 131066 filtered ports
PORT      STATE         SERVICE
1437/udp  open|filtered tabula
32445/udp open|filtered unknown
54395/udp open|filtered unknown
60114/udp open|filtered unknown

Nmap scan report for RPi3.lan (10.0.0.4)
Host is up (0.0030s latency).
All 131070 scanned ports on RPi3.lan (10.0.0.4) are filtered
    (127660) or open|filtered (3410)

Nmap scan report for RPiWired.lan (10.0.0.5)
Host is up (0.0037s latency).
All 131070 scanned ports on RPiWired.lan (10.0.0.5) are filtered
    (127505) or open|filtered (3565)

Nmap scan report for 10.0.0.6
Host is up (0.0036s latency).
All 131070 scanned ports on 10.0.0.5 are filtered (127692) or open|
    filtered (3378)

Nmap scan report for 10.0.0.7
Host is up (0.0017s latency).
All 131070 scanned ports on 10.0.0.7 are filtered (127714) or open|
    filtered (3356)

Nmap scan report for 10.0.0.8
Host is up (0.0066s latency).
All 131070 scanned ports on 10.0.0.8 are filtered (127880) or open|
    filtered (3190)

Nmap scan report for Win2.lan (10.0.0.9)
Host is up (0.0031s latency).
Not shown: 131065 filtered ports
```

```
PORT      STATE         SERVICE
1284/udp  open|filtered unknown
5026/udp  open|filtered unknown
20765/udp open|filtered unknown
22905/udp open|filtered unknown
34569/udp open|filtered unknown

Nmap scan report for 10.0.0.10
Host is up (0.0023s latency).
All 131070 scanned ports on 10.0.0.10 are filtered (127812) or open
    |filtered (3258)

Nmap scan report for 10.0.0.11
Host is up (0.0016s latency).
All 131070 scanned ports on 10.0.0.11 are filtered (127749) or open
    |filtered (3321)

Nmap scan report for 10.0.0.12
Host is up (0.0014s latency).
All 131070 scanned ports on 10.0.0.12 are filtered (127944) or open
    |filtered (3126)

Nmap scan report for 10.0.0.13
Host is up (0.0013s latency).
All 131070 scanned ports on 10.0.0.13 are filtered (127832) or open
    |filtered (3238)

Nmap scan report for 10.0.0.14
Host is up (0.0018s latency).
All 131070 scanned ports on 10.0.0.14 are filtered (127892) or open
    |filtered (3178)

Nmap scan report for 10.0.0.15
Host is up (0.0079s latency).
All 131070 scanned ports on 10.0.0.15 are filtered (127904) or open
    |filtered (3166)

Nmap scan report for dsldevice.lan (85.81.240.13)
Host is up (0.011s latency).
All 131070 scanned ports on dsldevice.lan (85.81.240.13) are
    filtered (127833) or open|filtered (3237)

Nmap scan report for dsldevice.lan (192.168.13.33)
Host is up (0.0016s latency).
Not shown: 131063 filtered ports
PORT      STATE         SERVICE
53/tcp    open          domain
53/udp    open          domain
67/udp    open|filtered dhcps
68/udp    open|filtered dhcpc
34603/udp open|filtered unknown
57343/udp open|filtered unknown
62112/udp open|filtered unknown
MAC Address: 30:91:8F:94:46:7A (Technicolor)

Nmap scan report for kali922.lan (192.168.13.42)
Host is up (0.000013s latency).
```

```
Not shown: 131065 closed ports
PORT      STATE         SERVICE
68/udp    open|filtered dhcpc
69/udp    open|filtered tftp
5353/udp  open|filtered zeroconf
27778/udp open|filtered unknown
50319/udp open|filtered unknown

Nmap done: 27 IP addresses (18 hosts up) scanned in 122969.52
    seconds
```

Listing E.4: Nmap TCP and UDP scan (done in one) from the hotspot network

E.0.5 TCP and UDP Scan to check for FTP services from the private network

```
root@kali922:~# nmap -Pn -sS -sU -p 1- 85.81.240.13 10.0.0.254

Starting Nmap 6.49BETA5 ( https://nmap.org ) at 2016-02-22 15:41
    CET
Nmap scan report for dsldevice.lan (85.81.240.13)
Host is up (0.0021s latency).
Not shown: 65535 open|filtered ports, 65525 filtered ports
PORT      STATE  SERVICE
21/tcp    open   ftp
53/tcp    open   domain
443/tcp   open   https
1723/tcp  open   pptp
2033/tcp  open   glogger
2034/tcp  open   scoremgr
2121/tcp  open   ccproxy-ftp
8000/tcp  open   http-alt
9001/tcp  closed tor-orport
23023/tcp open   unknown

Nmap scan report for Linux.lan (10.0.0.254)
Host is up (0.0027s latency).
Not shown: 131064 closed ports
PORT      STATE         SERVICE
21/tcp    open          ftp
139/tcp   open          netbios-ssn
445/tcp   open          microsoft-ds
41952/tcp open          unknown
137/udp   open          netbios-ns
138/udp   open|filtered netbios-dgm
MAC Address: 32:91:8F:94:46:7A (Unknown)

Nmap done: 2 IP addresses (2 hosts up) scanned in 65827.69 seconds
```

Listing E.5: Nmap TCP and UDP scan from the private network. The scan is done in order to check for open FTP services.

E.0.6 TCP and UDP Scan to check for FTP services from the hotspot network

```
root@kali922:~# nmap -Pn -sS -sU -p 1- 85.81.240.13 10.0.0.254

Starting Nmap 6.49BETA5 ( https://nmap.org ) at 2016-02-23 10:28
    CET
Nmap scan report for dsldevice.lan (85.81.240.13)
Host is up (0.0033s latency).
Not shown: 131065 filtered ports
PORT       STATE          SERVICE
21/tcp     open           ftp
67/udp     open|filtered dhcps
68/udp     open|filtered dhcpc
6984/udp   open|filtered unknown
59934/udp open|filtered unknown

Nmap scan report for Linux.lan (10.0.0.254)
Host is up (0.0021s latency).
All 131070 scanned ports on Linux.lan (10.0.0.254) are filtered

Nmap done: 2 IP addresses (2 hosts up) scanned in 50001.65 seconds
```

Listing E.6: Nmap TCP and UDP scan from the hotspot network. The scan is done in order to check for open FTP services.

APPENDIX

F

IP QOS CONFIGURATION

This Appendix includes:

Section F.1 on the following page: IP QoS labels before configuration.

Section F.2 on the next page: IP QoS rules before configuration.

Section F.3 on page 233: IP QoS labels after configuration.

Section F.4 on page 234: IP QoS rules after configuration.

Seciton D.3 on page 215 in Appendix D contains the list of default expressions.

F.1 IP QoS labels before configuration

```
=>label list
Id   Name         Class      Def    Ack         Fapp  Bid  Inh  Mark Type   Value   Use  Trace
----------------------------------------------------------------------------------------------
0    Interactive  increase   8      8           off   off  off  off  tos    0       14   off
1    Local        ignore     0      0           off   off  off  off  tos    0       2    off
2    Management   increase   12     12          on    off  off  off  tos    0       45   off
3    SIPS_RTP     overwrite  14     14          off   on   off  on   dscp   ef      1    off
4    SIPS_SIG     overwrite  12     12          off   on   off  on   dscp   af42    1    off
5    Video        increase   10     10          off   on   off  off  tos    0       3    off
6    VoIP-RTP     overwrite  14     14          on    on   off  on   dscp   ef      4    off
7    VoIP-Signal  overwrite  12     12          on    on   off  on   dscp   af42    10   off
8    default      increase   default
                                    prioritize  off   off  off  off  tos    0       1    off
9    voice-only   overwrite  14     14          off   off  on   off  tos    0       10   off
```

Listing F.1: The default IP QoS labels.

F.2 IP QoS rules before configuration

```
=>label rule list
Rules (flags: C=Constant, D=Dynamic, E=Enable, L=Log)
=====
Chain                 Nr.  Flags Rule name        Action                        Conditions
------------------------------------------------------------------------------------------------------------------------------------------

routing_labels        1    CDE                   : link rt_system_service       : _intf_loop.* > *.*
```

```
                      2    CDE                     : link rt_user_labels           : *.* > *.*
                      3    CDE                     : link rt_default_labels        : *.* > *.*
rt_user_labels        1    C E   voice-sip         : voice-only                    : sip local.* > *.*
                      2    C E   dhcp              : voice-only                    : dhcp local.* > *.*
qos_labels            1    CDE                     : link qos_system_service_in    : *.* > *._addr_local
                      2    CDE                     : link qos_system_service_out   : _intf_loop.* > *.*
                      3    CDE                     : link qos_user_labels          : *.* > *.*
                      4    CDE                     : link qos_default_labels       : *.* > *.*
qos_default_labels    1    C E   VoIP-Incomin...   : VoIP-Signal                   : VoIP-Inc-SIP-UDP wan.* > *.*
                      2    C E   VoIP-Incomin...   : VoIP-Signal                   : VoIP-Inc-SIP-TCP wan.* > *.*
                      3    C E   VoIP-Incomin...   : VoIP-RTP                      : VoIP-Inc-RTP wan.* > *.*
                      4    C E                     : Interactive                   : ah *.* > *.*
                      5    C E                     : Interactive                   : esp *.* > *.*
                      6    C E                     : Interactive                   : http *.* > *.*
                      7    C E                     : Interactive                   : httpproxy *.* > *.*
                      8    C E                     : Interactive                   : https *.* > *.*
                      9    C E                     : Interactive                   : imap *.* > *.*
                      10   C E                     : Interactive                   : imap3 *.* > *.*
                      11   C E                     : Interactive                   : imap4-ssl *.* > *.*
                      12   C E                     : Interactive                   : imaps *.* > *.*
                      13   C E                     : Interactive                   : pop2 *.* > *.*
                      14   C E                     : Interactive                   : pop3 *.* > *.*
                      15   C E                     : Interactive                   : pop3s *.* > *.*
                      16   C E                     : Interactive                   : smtp *.* > *.*
                      17   C E                     : Interactive                   : telnet *.* > *.*
                      18   C E                     : Management                    : dns *.* > *.*
                      19   C E                     : Management                    : icmp *.* > *.*
                      20   C E                     : Management                    : icmpv6 *.* > *.*
                      21   C E                     : Management                    : ike *.* > *.*
                      22   C E                     : Video                         : igmp *.* > *.*
                      23   C E                     : Video                         : rtsp *.* > *.*
                      24   C E                     : Local                         : local.* > *.*
                      25   C E   default           : default                       : *.* > *.*
```

```
rt_system_service      1     CDE                    : voice-only                    : VOIP_SIP_UDP_sv_out *.* >
                                                                                                            *._ipv4_addr
                       2     CDE                    : voice-only                    : VOIP_SIP_SIG_sv *.* > *._ipv4_addr
qos_system_service_in  1     CDE                    : Management                    : IGMP-Proxy_sv *._ipv4_addr > *.*
                       2     CDE                    : Management                    : DNS-S-UDP_sv_in *.* > *.*
                       3     CDE                    : Management                    : DNS-S-TCP_sv_in *.* > *.*
                       4     CDE                    : Management                    : DNS-S_sv *.* > *.*
                       5     CDE                    : Management                    : DNS-C_sv_in *._ipv4_addr > *.*
                       6     CDE                    : Management                    : DHCPv6-S_sv_in *._ipv6_addr > *.*
                       7     CDE                    : VoIP-Signal                   : VOIP_SIP_UDP_sv_in *._ipv4_addr >
                                                                                                                     *.*
                       8     CDE                    : VoIP-Signal                   : VOIP_SIP_SIG_sv *._ipv4_addr > *.*
                       9     CDE                    : Management                    : MLD_sv *._ipv6_addr > *.*
                       10    CDE                    : Management                    : NDSOL_sv *._ipv6_addr > *.*
                       11    CDE                    : Management                    : NDADV_sv *._ipv6_addr > *.*
                       12    CDE                    : Management                    : RTSOL_sv *._ipv6_addr > *.*
                       13    CDE                    : Management                    : RTADV_sv *._ipv6_addr > *.*
                       14    CDE                    : Management                    : DHCP-S_sv_in *._ipv4_addr > *.*
qos_system_service_out1      CDE                    : Management                    : IGMP-Proxy_sv *.* > *._ipv4_addr
                       2     CDE                    : Management                    : DNS-S-UDP_sv_out *.* > *.*
                       3     CDE                    : Management                    : DNS-S-TCP_sv_out *.* > *.*
                       4     CDE                    : Management                    : DNS-S_sv *.* > *.*
                       5     CDE                    : Management                    : DNS-C_sv_out *.* > *._ipv4_addr
                       6     CDE                    : Management                    : DHCPv6-S_sv_out *.* > *._ipv6_addr
                       7     CDE                    : VoIP-Signal                   : VOIP_SIP_UDP_sv_out *.* >
                                                                                                            *._ipv4_addr
                       8     CDE                    : VoIP-Signal                   : VOIP_SIP_SIG_sv *.* > *._ipv4_addr
                       9     CDE                    : Management                    : MLD_sv *.* > *._ipv6_addr
                       10    CDE                    : Management                    : NDSOL_sv *.* > *._ipv6_addr
                       11    CDE                    : Management                    : NDADV_sv *.* > *._ipv6_addr
                       12    CDE                    : Management                    : RTSOL_sv *.* > *._ipv6_addr
                       13    CDE                    : Management                    : RTADV_sv *.* > *._ipv6_addr
                       14    CDE                    : Management                    : DHCP-S_sv_out *.* > *._ipv4_addr
```

Listing F.2: The default IP QoS rules before the configuration.

F.3 IP QoS Labels after configuration

```
=>label list
Id   Name         Class      Def    Ack    Fapp  Bid   Inh  Mark Type   Value   Use   Trace
----------------------------------------------------------------------------------------
10   HotspotLabel overwrite  1      1      on    on    off  on   tos    0       1     off
0    Interactive  increase   8      8      off   off   off  off  tos    0       14    off
1    Local        ignore     0      0      off   off   off  off  tos    0       2     off
2    Management   increase   12     12     on    off   off  off  tos    0       45    off
3    SIPS_RTP     overwrite  14     14     off   on    off  on   dscp   ef      1     off
4    SIPS_SIG     overwrite  12     12     off   on    off  on   dscp   af42    1     off
5    Video        increase   10     10     off   on    off  off  tos    0       3     off
6    VoIP-RTP     overwrite  14     14     on    on    off  on   dscp   ef      4     off
7    VoIP-Signal  overwrite  12     12     on    on    off  on   dscp   af42    10    off
8    default      increase   6      6      off   off   off  off  tos    0       1     off
9    voice-only   overwrite  14     14     off   off   on   off  tos    0       10    off
```

Listing F.3: The IP QoS labels after the configuration. *HotspotLabel* has been added and *default* has been modified.

F.4 IP QoS rules after configuration

```
=>label rule list
Rules (flags: C=Constant, D=Dynamic, E=Enable, L=Log)
=====
Chain                 Nr.  Flags Rule name        Action                        Conditions
------------------------------------------------------------------------------------------------------------------------------------------

routing_labels        1    CDE                   : link rt_system_service       : _intf_loop.* > *.*
                      2    CDE                   : link rt_user_labels          : *.* > *.*
                      3    CDE                   : link rt_default_labels       : *.* > *.*
rt_user_labels        1    C E   voice-sip       : voice-only                   : sip local.* > *.*
                      2    C E   dhcp            : voice-only                   : dhcp local.* > *.*
qos_labels            1    CDE                   : link qos_system_service_in   : *.* > *._addr_local
                      2    CDE                   : link qos_system_service_out  : _intf_loop.* > *.*
                      3    CDE                   : link qos_user_labels         : *.* > *.*
                      4    CDE                   : link qos_default_labels      : *.* > *.*
qos_user_labels       1    C E   HotspotQoS      : HotspotLabel                 : dmz.* > *.*
qos_default_labels    1    C E   VoIP-Incomin... : VoIP-Signal                  : VoIP-Inc-SIP-UDP wan.* > *.*
                      2    C E   VoIP-Incomin... : VoIP-Signal                  : VoIP-Inc-SIP-TCP wan.* > *.*
                      3    C E   VoIP-Incomin... : VoIP-RTP                     : VoIP-Inc-RTP wan.* > *.*
                      4    C E                   : Interactive                  : ah *.* > *.*
                      5    C E                   : Interactive                  : esp *.* > *.*
                      6    C E                   : Interactive                  : http *.* > *.*
                      7    C E                   : Interactive                  : httpproxy *.* > *.*
                      8    C E                   : Interactive                  : https *.* > *.*
                      9    C E                   : Interactive                  : imap *.* > *.*
                      10   C E                   : Interactive                  : imap3 *.* > *.*
                      11   C E                   : Interactive                  : imap4-ssl *.* > *.*
                      12   C E                   : Interactive                  : imaps *.* > *.*
                      13   C E                   : Interactive                  : pop2 *.* > *.*
                      14   C E                   : Interactive                  : pop3 *.* > *.*
                      15   C E                : Interactive                  : pop3s *.* > *.*
                      16   C E                : Interactive                  : smtp *.* > *.*
                      17   C E                : Interactive                  : telnet *.* > *.*
                      18   C E                : Management                   : dns *.* > *.*
                      19   C E                : Management                   : icmp *.* > *.*
                      20   C E                : Management                   : icmpv6 *.* > *.*
                      21   C E                : Management                   : ike *.* > *.*
                      22   C E                : Video                        : igmp *.* > *.*
                      23   C E                : Video                        : rtsp *.* > *.*
                      24   C E                : Local                        : local.* > *.*
                      25   C E   default      : default                      : *.* > *.*
rt_system_service     1    CDE                : voice-only                   : VOIP_SIP_UDP_sv_out *.* >
                                                                                                      *._ipv4_addr
                      2    CDE                : voice-only                   : VOIP_SIP_SIG_sv *.* > *._ipv4_addr
qos_system_service_in 1    CDE                : Management                   : IGMP-Proxy_sv *._ipv4_addr > *.*
                      2    CDE                : Management                   : DNS-S-UDP_sv_in *.* > *.*
                      3    CDE                : Management                   : DNS-S-TCP_sv_in *.* > *.*
                      4    CDE                : Management                   : DNS-S_sv *.* > *.*
                      5    CDE                : Management                   : DNS-C_sv_in *._ipv4_addr > *.*
                      6    CDE                : Management                   : DHCPv6-S_sv_in *._ipv6_addr > *.*
                      7    CDE                : VoIP-Signal                  : VOIP_SIP_UDP_sv_in *._ipv4_addr >
                                                                                                               *.*
                      8    CDE                : VoIP-Signal                  : VOIP_SIP_SIG_sv *._ipv4_addr > *.*
                      9    CDE                : Management                   : MLD_sv *._ipv6_addr > *.*
                      10   CDE                : Management                   : NDSOL_sv *._ipv6_addr > *.*
                      11   CDE                : Management                   : NDADV_sv *._ipv6_addr > *.*
                      12   CDE                : Management                   : RTSOL_sv *._ipv6_addr > *.*
                      13   CDE                : Management                   : RTADV_sv *._ipv6_addr > *.*
                      14   CDE                : Management                   : DHCP-S_sv_in *._ipv4_addr > *.*
qos_system_service_out1    CDE                : Management                   : IGMP-Proxy_sv *.* > *._ipv4_addr
                      2    CDE                : Management                   : DNS-S-UDP_sv_out *.* > *.*
                      3    CDE                : Management                   : DNS-S-TCP_sv_out *.* > *.*
                      4    CDE                : Management                   : DNS-S_sv *.* > *.*
```

```
5    CDE                  : Management                    : DNS-C_sv_out *.* > *._ipv4_addr
6    CDE                  : Management                    : DHCPv6-S_sv_out *.* > *._ipv6_addr
7    CDE                  : VoIP-Signal                   : VOIP_SIP_UDP_sv_out *.* >
                                                                                   *._ipv4_addr
8    CDE                  : VoIP-Signal                   : VOIP_SIP_SIG_sv *.* > *._ipv4_addr
9    CDE                  : Management                    : MLD_sv *.* > *._ipv6_addr
10   CDE                  : Management                    : NDSOL_sv *.* > *._ipv6_addr
11   CDE                  : Management                    : NDADV_sv *.* > *._ipv6_addr
12   CDE                  : Management                    : RTSOL_sv *.* > *._ipv6_addr
13   CDE                  : Management                    : RTADV_sv *.* > *._ipv6_addr
14   CDE                  : Management                    : DHCP-S_sv_out *.* > *._ipv4_addr
```

Listing F.4: The IP QoS rules after the configuration. *HotspotQoS* has been added in chain *qos_user_labels*

APPENDIX

G

QOS TEST PROGRAM

This chapter will describe the program designed for the QoS bandwidth measurements used throughout the report. The program is designed to be dynamic in terms of different test setups in terms of measurement times and delays. A client is given individual measurement times and delays that makes it possible to create test setups close to real world situations. Throughput measurements are typically done by sending as much data as fast as possible and then measure the amount of data received over time[1] [Mandrup et al. [2014]].

The purpose of the QoS test program is to measure and plot real-time throughput changes, instead of only giving a single average throughput value. This allows for insight into how the throughput changes over time which will create results that can accurately show how the QoS implementation performs in a changing environment. Figure G.1 shows an example of a throughput test (the setup is not important), that shows the real-time throughput and the total average throughput.

To visualize how the program is designed, Figure G.2 shows a sequence diagram of an upload bandwidth performance test with one client. Green arrows indicates network traffic, and red arrows indicates specific component operations.

The listening server consists of a TCP socket server that establishes a connection to connecting clients. When the connection is established, the server creates and forwards the socket connection to a new thread specific for that client. For every client that connects a thread is created to control the communication and later perform the test. This is to ensure that the tests are done in parallel. The clients side will transmit its host name when a connection is established, which allows the thread to transmit test-specific-variables to the client. The specific variables consists of a port number, total

[1] There are other methods to measure network throughput that uses less data, but for the scope of this project a brute-force measurement method is adequate.

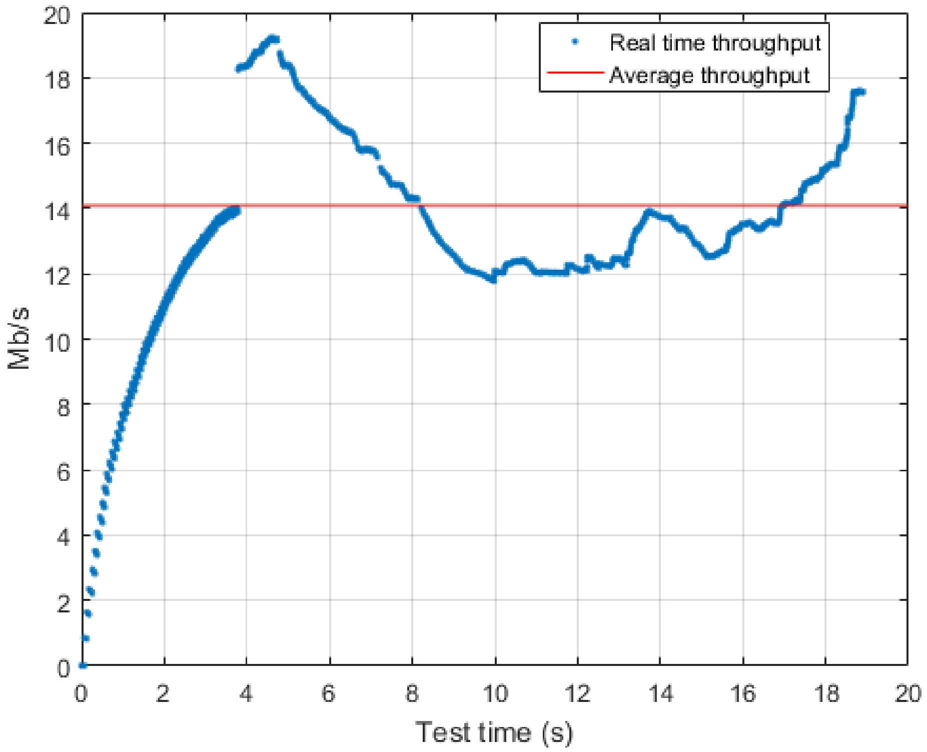


Figure G.1: An example of a throughput test that shows the real-time throughput as well as the total average throughput

test time and a delay for test start. The port will be used as a destination port for the performance test, and is different for every client. The test time and delay is manually set for each client before the server is started. The variables are used to easily design different test setups.

When the connections has been established, and necessary information is exchanged, each client is ready to perform the test. Before running, each client will wait for a start signal to ensure that tests starts in sync. The start signal is triggered by a user input, such that the tests start when all desired clients have connected and is ready. Triggering the start test will cause the thread on the server to bind a datagram socket to the specific port transmitted to the client, which will be used for receiving the UDP transmitted data for measuring throughput. When the socket is ready it will transmit the start signal to the client and both the client and the server thread will, after the delay, enter a loop that runs for the specified test time. The client will be transmitting as much data as possible, and the server analyses the throughput continuously creating real time results. Equation G.1 is used to calculate the average application layer throughput for each received packet on the server [Mandrup et al. [2014]]:

$$Throughput = \frac{\frac{P \cdot i \cdot 8}{1048576}}{\Delta t} \text{[Mbps]} \tag{G.1}$$

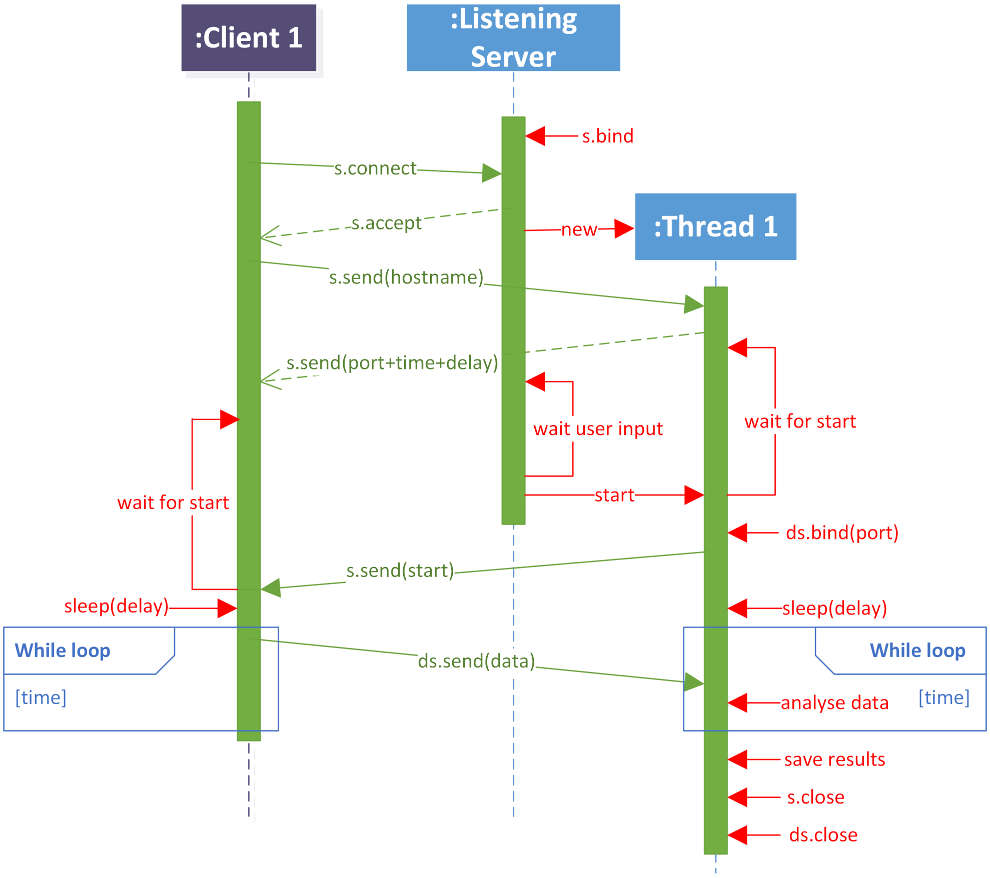


Figure G.2: Sequence diagram of one client performing an upload test.

P	Packet size (application layer)	[Bytes]
i	Amount of packets received	[$\cdot$]
t	Time spent receiving i packets	[s]
8	Transform bytes into bits	[$\cdot$]
1048576	Transform into Mb (1 mebibit $= 2^{20}$ bits)	[$\cdot$]

In an effort to see rapid changes in throughput and avoid the results depending too much on previous packets, the equation will only consider the most recent 8000 packets. This means that i will be a maximum of 8000 packets and Δt will consider the time spent receiving the most recent $i = 8000$ packets.

Finally, when a test has ended, the client will exit the while loop and close down. The server saves the result in a *.txt*-file and closes the thread.

APPENDIX

H

FINAL CONFIGURATION OF THE MAG

This Appendix includes:

Section H.1 : **CLI commands for Network Integrity**

Section H.2 : **CLI commands for Quality of Service**

Section H.3 : **CLI commands for Authentication and User Identification**

The full list of unmodified configurations can be found in *ListOfCLICommands.txt* on the group server: `http://kom.aau.dk/group/16gr1022/`.

H.1 CLI commands for Network Integrity

Listing H.1 shows the full list of CLI commands for Network Integrity (Part I).

```
=> wireless mssid ifadd ssid=TelenorHotspot
=> wireless mssid ifconfig ssid_id=1 ssid=TelenorHotspot
   apisolation=enabled any=enabled secmode=wpa WPAradiuskey="
   radiusSecret" WPAradiusip=192.38.55.78 WPAradiusport=20012
   WPAPSKversion=WPA2 WPAversion=WPA2 radio_id=0 maxassociations=8
=> wireless mssid ifattach ssid_id=1

=> eth bridge ifadd intf=HotspotBridge dest=wl_ssid1_local0
=> eth bridge ifattach intf=HotspotBridge
=> eth bridge config vlan=enabled
=> eth vlan add name=HotspotVLAN vid=3
=> eth bridge vlan ifadd name=HotspotVLAN intf=OBC untagged=
   disabled
=> eth bridge vlan ifadd name=HotspotVLAN intf=HotspotBridge
   untagged=enabled
```

```
=> eth bridge vlan ifdelete name=default intf=HotspotBridge

=> eth ifadd intf=HotspotEth
=> eth ifconfig intf=HotspotEth dest=bridge vlan=HotspotVLAN
=> eth ifattach intf=HotspotEth

=> ip ifadd intf=HotSpotIP dest=HotspotEth
=> ip ifconfig intf=HotSpotIP group=dmz ipv6=disabled
=> ip ifattach intf=HotSpotIP
=> ip ipadd intf=HotSpotIP addr=192.168.13.33 netmask=28
=> ip ipconfig addr=192.168.13.33 preferred=enabled primary=enabled
=> nat ifconfig intf=HotSpotIP translation=transparent
=> dhcp server pool add name=HotspotDHCP
=> dhcp server pool config name=HotspotDHCP intf=HotSpotIP
   poolstart=192.168.13.37 poolend=192.168.13.44 netmask=28
   gateway=192.168.13.33 leasetime=300 primdns=212.242.40.3 secdns
   =212.242.40.51 localdns=disabled
=> dhcp relay ifconfig intf=HotSpotIP relay=enabled
=> dhcp relay add name=HotspotRelay
=> dhcp relay modify name=HotspotRelay addr=127.0.0.1 intf=
   HotSpotIP giaddr=192.168.13.33

=> firewall rule add chain=forward_level_Standard index=1 name=
   HStoWAN srcintf=dmz dstintf=wan state=enabled action=accept
=> firewall rule add chain=forward_level_Standard index=2 name=
   HStoALL srcintf=dmz state=enabled action=deny
=> firewall rule add chain=forward_level_Standard index=3 name=
   ALLtoHS dstintf=dmz state=enabled action=deny

=> firewall rule add chain=sink_system_service index=1 name=
   HSgateway srcintf=lan dstip=192.168.13.33 log=disabled state=
   enabled action=deny
=> firewall rule add chain=sink_system_service index=2 name=dmzicmp
    srcintf=dmz dstip=192.168.13.33 serv=icmp log=disabled state=
   enabled action=accept
=> firewall rule add chain=sink_system_service index=3 name=dmzdns
   srcintf=dmz dstip=192.168.13.33 serv=dns log=disabled state=
   enabled action=accept
=> firewall rule add chain=sink_system_service index=4 name=dmzdhcp
    srcintf=dmz serv=dhcp log=disabled state=enabled action=accept
=> firewall rule add chain=sink_system_service index=5 name=dmzall
   srcintf=dmz log=disabled state=enabled action=deny
```

Listing H.1: The full list of CLI commands for Network Integrity (Part I).

H.2 CLI commands for Quality of Service

Listing H.2 shows the full list of CLI commands for Quality of Service (Part II). As the list is too long to print X takes the values *37* to *44*.

```
=> label add name=HotspotLabel
=> label modify name=HotspotLabel classification=overwrite defclass
   =1 ackclass=1 bidirectional=enabled inheritance=disabled
```

```
    tosmarking=disabled fapprio=enabled
=> label rule add chain=qos_user_labels index=1 name=HotspotQoS
    srcintf=dmz log=disabled state=enabled label=HotspotLabel
=> label modify name=default defclass=6 ackclass=6

=> eth bridge ifconfig intf=HotspotBridge priority=1
=> eth bridge ifconfig intf=HotspotBridge prioconfig=overwrite

=> nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=192.168.13.X outside_port=5000-9998 inside_port
    =1-4999 mode=outbound weight=10 status=up description=HSXRule1
=> nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=192.168.13.X outside_port=5000-26847 inside_port
    =5000-26847 mode=outbound weight=10 status=up description=HS
    XRule2
=> nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=192.168.13.X outside_port=5000-24343 inside_port
    =26848-46191 mode=outbound weight=10 status=up description=HS
    XRule3
=> nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=192.168.13.X outside_port=7504-26847 inside_port
    =46192-65535 mode=outbound weight=10 status=up description=HS
    XRule4
```

Listing H.2: The full list of CLI commands for Quality of Service (Part II). X takes the values *37* to *44*.

H.3 CLI commands for Authentication and User Identification

The list of CLI commands for Authentication and User Identification (Part III) is simply too long to print as 192 NAPT rules is created. Listing H.3 shows a "*default*" template for NAPT rules, where X is a variable that specifies the IP address of the hotspot user, which can take the values `[37-44]`. Each user needs 24 rules as shown in Table H.1 in order to map all possible **inside_port** ports. The variable Y specifies the number of the rule. The variables wwwww, and yyyyy specifies the dedicated **inside_port** port range for each rule also according to Table H.1. The variables zzzzz, qqqqq is used to specify the dedicated **outside_port** port range for each hotspot user according to Table H.2.

```
=> nat tmpladd intf=internet_pppoe type=napt outside_addr=0.0.0.1
    inside_addr=192.168.13.X outside_port=zzzzz-qqqqq inside_port=
    wwwww-yyyyy mode=outbound weight=10 status=up description=User
    XRuleY

=> wireless radius server config radio_id=0 ssid_id=1 radius_id=0
    state=enabled ip=192.38.55.78 port=20013 dhcpaccounting=yes
    type=accounting secret=radiusSecret
```

Listing H.3: The list of CLI commands for Authentication and User Identification (Part

III). Line 1 specifies a "*default*" setup for NAPT rules.

Rule	Dedicated port range (**inside_port**)
1	1-2731
2	2732-5462
3	5463-8193
4	8194-10924
5	10925-13655
6	13656-16386
7	16387-19117
8	19118-21848
9	21849-24579
10	24580-27310
11	27311-30041
12	30042-32772
13	32773-35503
14	35504-38234
15	38235-40965
16	40966-43696
17	43697-46427
18	46428-49158
19	49159-51889
20	51890-54620
21	54621-57351
22	57352-60082
23	60083-62813
24	62814-65535

Table H.1: **inside_port** port range for each rule.

User	IP address	Dedicated port range (**outside_port**)
1	192.168.13.37	5000-7730
2	192.168.13.38	7731-10461
3	192.168.13.39	10462-13192
4	192.168.13.40	13193-15923
5	192.168.13.41	15924-18654
6	192.168.13.42	18655-21385
7	192.168.13.43	21386-24116
8	192.168.13.44	24117-26847

Table H.2: Dedicated **outside_port** port range for each hotspot user.

In order to show how the list can be generated, Listing H.4 is shown. The Listing shows how MATLAB can be used to print out all NAPT rules. The following definitions has been made: *numusers=8*, *numrules=24*, and the *outsiderange* and *insiderange* matrixes are defined according to Tables H.1 and H.2.

```
numusers = 8;
numrules = 24;

insiderange = [1,2731; 2732,5462; 5463,8193; 8194,10924;
    10925,13655; 13656,16386; 16387,19117; 19118,21848;
    21849,24579; 24580,27310; 27311,30041; 30042,32772;
    32773,35503; 35504,38234; 38235,40965; 40966,43696;
    43697,46427; 46428,49158; 49159,51889; 51890,54620;
    54621,57351; 57352,60082; 60083,62813; 62814,65535];

outsiderange = [5000,7730; 7731,10461; 10462,13192; 13193,15923;
    15924,18654; 18655,21385; 21386,24116; 24117,26847];

for user = 1:numusers
    for rule = 1:numrules
            str = sprintf('nat tmpladd intf=internet_pppoe type=
                napt outside_addr=0.0.0.1 inside_addr=192.168.13.%d
                 outside_port=%d-%d inside_port=%d-%d mode=outbound
                 weight=10 status=up description=User%dRule%d',(
                user+36),outsiderange(user ,1),outsiderange(user ,2),
                insiderange(rule ,1),insiderange(rule ,2),(user+36),
                rule);
            disp(str)
    end
end
```

Listing H.4: An extracted script from a piece of MATLAB code that creates NAPT rules.

BIBLIOGRAPHY

Architecture and Group, **2006**. Architecture and Transport Working Group. *Migration to Ethernet-Based DSL Aggregation* . Technical Report - DSL Forum, 2006. [Online; accessed 23-November-2015].

Benny, **2011**. Benny. *Find computer names on a private network (with nmap?)*. superuser question - `http://superuser.com/questions/358156/find-computer-names-on-a-private-network-with-nmap`, 2011. [Online; accessed 14-December-2015].

Bernard Aboba, **September 2003**. Pat R. Calhoun Bernard Aboba. *RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP)*, Internet Engineering Task Force, September 2003. URL `https://tools.ietf.org/html/rfc3579`.

Bull and Matthew, **2015**. DEF CON Ronny Bull and Jeanna Matthew. *DEF CON 23 - Ronny Bull and Jeanna Matthews -Exploring Layer 2 Security in Virtualized Environments*. `https://www.youtube.com/watch?v=RaA5dEIqzzQ`, 2015. [Online; accessed 04-March-2016].

Carl Rigney, Ward Willats, **June 2000a**. Bernard Aboba Carl Rigney, Ward Willats. *RADIUS Extensions*, Internet Engineering Task Force, June 2000a. URL `https://www.ietf.org/rfc/rfc2869.txt`.

Carl Rigney, Allan C. Rubens, **June 2000b**. William Allen Simpson Steve Willens Carl Rigney, Allan C. Rubens. *Remote Authentication Dial In User Service (RADIUS)*, Internet Engineering Task Force, June 2000b. URL `https://tools.ietf.org/html/rfc2865`.

Carl Rigney, Allan C. Rubens, **July 1993**. William Allen Simpson Steve Willens Carl Rigney, Allan C. Rubens. *An Access Control Protocol, Sometimes Called TACACS*, Internet Engineering Task Force, July 1993. URL `https://tools.ietf.org/html/rfc1492`.

Cisco, **2016**. Cisco. *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper*. `http:`

//www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html, 2016. [Online; accessed 22-March-2016].

Cisco, **2006**. Cisco. *Understanding the Ping and Traceroute Commands*. http://www.cisco.com/c/en/us/support/docs/ios-nx-os-software/ios-software-releases-121-mainline/12778-ping-traceroute.html, 2006. [Online; accessed 27-November-2015].

D. Carrel, **January 1997**. Lol Grant D. Carrel. *The TACACS+ Protocol Version 1.78*, Internet Engineering Task Force, January 1997. URL https://tools.ietf.org/html/draft-grant-tacacs-02.

Decker, **2014**. Pieter De Decker. *Inside the Domain Name System*. https://www.youtube.com/watch?v=GlZC4Jwf3xQ, 2014. [Online; accessed 16-November-2015].

Encyclopedia, **2016**. The Network Encyclopedia. *Dynamic Host Configuration Protocol (DHCP)*. http://www.thenetworkencyclopedia.com/entry/dynamic-host-configuration-protocol-dhcp/, 2016. [Online; accessed 17-February-2016].

Eric Kornum, **2016**. Version2 Eric Kornum. *Ericsson klar med 5G i 2020*. http://www.version2.dk/artikel/ericsson-klar-med-5g-i-2020-633186, 2016. [Online; accessed 23-March-2016].

Ericson, **2015a**. Ericson. *ERICSSON MOBILITY REPORT - Europe*. http://www.ericsson.com/res/docs/2015/mobility-report/emr-nov-2015-regional-report-europe.pdf, 2015. [Online; accessed 22-March-2016].

Ericson, **2015b**. Ericson. *ERICSSON MOBILITY REPORT*. http://www.mypresswire.com/log/pm_files/file_31687.pdf, 2015. [Online; accessed 22-March-2016].

hacker, **2013a**. ethical hacker. *Configure Free-radius-Server on Kali Linux Part 1*. https://www.youtube.com/watch?v=wMdZiOHiNpA, 2013. [Online; accessed 23-November-2015].

hacker, **2013b**. ethical hacker. *FreeRadius Server Configuration PART 2*. https://www.youtube.com/watch?v=ijyiN-urD7g, 2013. [Online; accessed 23-November-2015].

freesoft.org, **2015**. freesoft.org. *The in-addr.arpa Domain*. http://www.freesoft.org/CIE/Course/Section2/15.htm, 2015. [Online; accessed 15-December-2015].

Geier, **2011**. Eric Geier. *5 Free RADIUS Testing and Monitoring Tools*. http://www.serverwatch.com/server-reviews/article.php/3935211/5-Free-RADIUS-Testing-and-Monitoring-Tools.htm, 2011. [Online; accessed 24-November-2015].

Glen Zorn, Dave Mitton, June 2000. Bernard Aboba Glen Zorn, Dave Mitton. *RADIUS Accounting Modifications for Tunnel Protocol Support*, Internet Engineering Task Force, June 2000. URL https://tools.ietf.org/html/rfc2867.

IANA, 2016. IANA. *Service Name and Transport Protocol Port Number Registry*. http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml, 2016. [Online; accessed 01-Mai-2016].

IANA, 2015. IANA. *Assigned Internet Protocol Numbers*. http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml, 2015. [Online; accessed 16-December-2015].

(IANA), 2015. The Internet Assigned Numbers Authority (IANA). *IANA IPv4 Address Space Registry*. http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml, 2015. [Online; accessed 16-November-2015].

itgeared, 2012. itgeared. *DHCP Relay Agent Overview*. https://www.youtube.com/watch?v=jPfZJNRJYM0, 2012. [Online; accessed 01-May-2016].

John Blackford, 2013. Mike Digdon John Blackford. *TR-069 CPE WAN Management Protocol - Amendment 5*. https://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf, 2013. [Online; accessed 01-May-2016].

Juniper Networks, 2014. Inc. Juniper Networks. *Implementation guide: Broadband edge solution. BroadbandEdgeSolution[8010092-en].pdf*, 2014. [Online; accessed 23-November-2015].

Juniper Networks, Jeremy Schulman, 2011. Lenny Pollard John Rolfe Juniper Networks, Jeremy Schulman. *Day one: Dynamic subscriber management*. Day One. Juniper Networks, 2011. ISBN 978-1-936779-43-7.

Mandrup, Hoejholt, and Thomsen, 2014. Daniel Mandrup, Nikolaj Bové Hoejholt, and Jesper Graversgaard Thomsen. *Project paper - Performance Evaluation of Methods for Estimating Bandwidth on Cellular Connections.pdf*. Performance_Evaluation_of_Methods_for_Estimating_Bandwidth_on_Cellular_Connections.pdf, 2014. [Location: Group server http://kom.aau.dk/group/16gr1022/].

Microsoft, 2016a. Microsoft. *DHCP (Dynamic Host Configuration Protocol) Basics*. https://support.microsoft.com/en-us/kb/169289, 2016. [Online; accessed 17-February-2016].

Microsoft, 2016b. Microsoft. *Lease Renewals*. https://technet.microsoft.com/en-us/library/cc958919.aspx, 2016. [Online; accessed 17-February-2016].

Mitton, **July 2000a**. David Mitton. *Network Access Servers Requirements:Extended RADIUS Practices*, Internet Engineering Task Force, July 2000a. URL `https://www.ietf.org/rfc/rfc2882.txt`.

Mitton, **July 2000b**. David Mitton. *Network Access Server Requirements Next Generation (NASREQNG) NAS Model*, Internet Engineering Task Force, July 2000b. URL `https://tools.ietf.org/html/rfc2881`.

Murtagh, **2012**. Fionn Murtagh. *Moore's Law, the Rise of the Data Economy, and the Problem of "Dirty Data"*. `https://fionnmurtagh.wordpress.com/2012/05/06/moores-law-the-rise-of-the-data-economy-and-the-problem-of-dirty-data/`, 2012. [Online; accessed 23-March-2016].

Nmap, **2015a**. Nmap. *Host Discovery*. `https://nmap.org/book/man-host-discovery.html`, 2015. [Online; accessed 14-December-2015].

Nmap, **2015b**. Nmap. *Introduction*. `https://nmap.org/`, 2015. [Online; accessed 26-November-2015].

Nmap, **2015c**. Nmap. *Nmap Reference Guide*. `https://nmap.org/book/man.html`, 2015. [Online; accessed 10-December-2015].

Nmap, **2015d**. Nmap. *Port Scanning Basics*. `https://nmap.org/book/man-port-scanning-basics.html`, 2015. [Online; accessed 01-December-2015].

Notenboom, **2013**. Leo Notenboom. *Is my router acting as a DNS server?* `https://askleo.com/is-my-router-acting-as-a-dns-server/`, 2013. [Online; accessed 16-November-2015].

notes, **2016**. Shichao's notes. *User Datagram Protocol (UDP) and IP Fragmentation*. `https://notes.shichao.io/tcpv1/ch10/`, 2016. [Online; accessed 24-May-2016].

NPR, **2015**. NPR. *Advanced DMZ Network on a Thomson Router*. `http://npr.me.uk/advdmz.html`, 2015. [Online; accessed 16-November-2015].

Rigney, **June 2000**. Carl Rigney. *RADIUS Accounting*, Internet Engineering Task Force, June 2000. URL `https://tools.ietf.org/html/rfc2866`.

Rouse, **2013**. Margaret Rouse. *femtocell*. `http://searchtelecom.techtarget.com/definition/femtocell`, 2013. [Online; accessed 24-March-2016].

S. Wadhwa, J. Moisand, **2011**. T. Haag N. Voigt T. Taylor S. Wadhwa, J. Moisand. *Protocol for Access Node Control Mechanism in Broadband Networks*. RFC 6320 - Internet Engineering Task Force (IETF), 2011. [Online; accessed 23-November-2015].

Sandal, 2014. Version2 Jesper Stein Sandal. *YouSee lukker for hotspots på grund af sikkerhedshul*. http://www.version2.dk/artikel/yousee-lukker-hotspots-paa-grund-af-sikkerhedshul-57044, 2014. [Online; accessed 16-November-2015].

SAVVIUS, 2016. Wildpackets SAVVIUS. *IP Type Of Service*. http://www.wildpackets.com/resources/compendium/tcp_ip/ip_tos, 2016. [Online; accessed 29-February-2016].

Society, 2005. IEEE Computer Society. *IEEE Std 802.11e-2005 (Amendment to IEEE Std 802.11, 1999 Edition (Reaff 2003)*. IEEE standard, 2005.

Stahl, 2015. Emil Stahl. *Internetudbydere og IPv6*. https://ipv6-adresse.dk/, 2015. [Online; accessed 14-December-2015].

Sørensen, 2016. Troels Bundgaard Sørensen. *Wireless Systems Performance mm5 - Cell splitting and Sectoring*. Lecture in Wireless System Performance, 2016.

Technicolor, 2012a. Technicolor. *CLI REFERENCE GUIDE R10.4 . CLI Guide TG799vn R10.4 [218520].pdf*, 2012.

Technicolor, 2012b. Technicolor. *Config guides*. http://support.alcadis.nl/downloads/Technicolor/General/General%20Guides/, 2012. [Online; accessed 16-November-2015].

Technicolor, 2008. Technicolor. *Thomson Gateways and Multiple IP Addresses*. http://support.alcadis.nl/downloads/Technicolor/General/General%20Guides/, 2008. [Online; accessed 16-November-2015].

Telstra, 2016. Telstra. *THE TELSTRA AIR NETWORK*. https://www.telstra.com.au/broadband/telstra-air/how-it-works, 2016. [Online; accessed 24-March-2016].

The Texas Higher Education Network, 2011. managed by the UT System Office of Telecommunication Services The Texas Higher Education Network. *Reverse DNS Lookup*. https://www.the.net/tools/docs/reverse.php, 2011. [Online; accessed 15-December-2015].

Tofel, 2013. Kevin C. Tofel. *Who has the largest Wi-Fi network in the US? Cable companies say they do*. https://gigaom.com/2013/06/10/who-has-the-largest-wi-fi-network-in-the-us-cable-companies-say-they-do/, 2013. [Online; accessed 24-March-2016].

Walt, 2011. Dirk van der Walt. *FreeRADIUS Beginner's Guide: Manage your network resources with FreeRADIUS*. Packt Publishing, 2011. ISBN 978-1-849514-08-8.

Version2, 2014. Lasse Hedegaard Poulsen Version2. *Naboens signalforstærker giver dig dårlig dækning - og er pivulovlig*. http://www.version2.dk/artikel/naboens-signalforstaerker-giver-dig-daarlig-daekning-og-er-pivulovlig-57647, 2014. [Online; accessed 24-March-2016].

Victor Fajardo, Jari Arkko, **October 2012**. John Loughney Glen Zorn Victor Fajardo, Jari Arkko. *Diameter Base Protocol*, Internet Engineering Task Force, October 2012. URL `https://tools.ietf.org/html/rfc6733`.

Vince Mammoliti, Glen Zorn, **September 2006**. Peter Arberg Vince Mammoliti, Glen Zorn. *DSL Forum Vendor-Specific RADIUS Attributes*, Internet Engineering Task Force, September 2006. URL `https://tools.ietf.org/html/rfc4679`.

Wikipedia, **2015**. Wikipedia. *DHCP relaying*. `https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol#DHCP_relaying`, 2015. [Online; accessed 16-November-2015].