POLITECNICO DI TORINO

Master's Degree in computer engineering

Master's Degree Thesis

Next generation license-free wireless regional and metropolitan area networks. Layer 3 migration for an effective exploitation of layer 1 performance

Supervisors

Prof. Daniele TRINCHERO

Ing. Giovanni COLUCCI

Candidate

Simone PANIATI

December 2023

Summary

In rural areas of Italy, a stable and fast internet connection is not always available, due to low population density, which makes economically unviable for traditional internet service providers to build a modern infrastructure based on optical fiber or equivalent technologies. The mission of the association "Senza Fili Senza Confini", which is the first non-profit WISP in Italy, is to provide a fast and reliable internet connection to its associates by using wireless technologies. Since the foundation of the association in 2014, its network has undergone a constant growth: starting from a few nodes, nowadays the network counts about 310 nodes and 8500 regular users. During these nine years new IEEE80211 standards have appeared. The original network configuration was not scalable enough to adapt to the new standards and topology, with the need to provide advanced functionalities like redundancy, traffic balancing and automatic configuration. The goal of this thesis project is to create a new network configuration tailored to the new available technologies and capable of providing all the advanced functionalities required in a modern wireless network. As a result, the bandwidth capacity of each single connection has been increased, from original 12 Mb/s to current 100 Mb/s.

Table of Contents

List of Figures

Acronyms

SFSC
: senza Fili Senza Confini

WISP
: wireless internet service provider

ADSL
: advanced digital subscriber line

VLAN
: virtual local area network

OSI
: open systems interconnection

ISP
: internet service provider

SO-HO
: small office / home office

CPE
: Customer premise equipment

ASIC
: application specific integrated circuit

NAT
: network address translation

BGP

Border gateway protocol

OSPF

Open shortest path first

AS

Autonomous system

IGP

Interior gateway protocol

EGP

Exterior gateway protocol

MPLS

Multiprotocol label switching

PoE

Power over ethernet

VRF

Virtual routing and forwarding

VPN

Virtual private network

QoS

Quality of service

PPPoE

Point to Point protocol over Ethernet

CGNAT

Carrier grade network address translation

ECMP

Equal cost multipath routing

CPU

Central processing unit

MAC

Medium Access Control

TDMA

Time division multiple access

Chapter 1

Introduction

1.1 The association "Senza Fili Senza Confini"

Italian ISPs has always struggled to provide fast and reliable internet connections in rural and sparsely populated areas like countrysides, mountains and hills, where the geography of the territory makes building a modern internet infrastructure based on optical fibers difficult and not economically viable. The inhabitants of these areas had to rely on advanced digital subscriber line (ADSL) connections which use the existing telephone network, but this approach presents some problems:

- Insufficient bandwidth for modern internet services (video/music streaming, videoconferencing, smart-working)
- Unreliable connections
- Expensive subscriptions

The non-profit association senza Fili Senza Confini (SFSC) [1], which operates as a wireless internet service provider (WISP) in Piedmont, has successfully solved some of these issues by building a geographical wireless network in order to provide a cheap, fast and reliable internet connection in rural and digital-divided areas. SFSC was born in 2014 in the small municipality of Verrua Savoia after an experimental project which lasted about 52 months with the aim of verifying the possibility of bringing low-cost broadband in rural areas. Nowadays, 9 years later, SFSC operates in more than 100 municipalities and has over 8200 associates that can benefit of a fast internet connection with unlimited traffic available. Associates don't just benefit of a cheap and reliable internet connection because the main objective of SFSC is the reduction of digital divide and the internet connection is just one of the means to achieve this goal: seminars and courses about the digital word are frequently organized for free, especially for children and elderly people.

1.2 Research objectives

Since the network of SFSC has undergone a constant growth in the past years, the complexity of the network increased as well and the requirements have changed. The existing network architecture relies on virtual local area network (VLAN) and ISO-OSI layer 2 forwarding in order to deliver packets across the network, which poses a series of limitations that will be described in details in chapter 5. After identifying the limitations and new requirements of the network, the candidate analyzed and tested the available technologies and protocols in order to create a new network configuration based on IP routing and dynamic routing protocols that is capable of satisfying the project requirements. The thesis project developed according to the following points:

- Identification of the current network limitations and new requirements
- Identification and analysis of the available network technologies and protocols
- Initial tests and simulations of small networks to identify the best technologies and configurations that are able to satisfy the needs of the project
- Deployment and testing of the new configuration in a small section of the existing network
- Deployment of the new configuration in the complete network
- Verification and testing of the new configuration

The thesis project is concluded by verifying that all the project requirements were satisfied and by presenting improvement opportunities.

Chapter 2

Computer networks overview

In this chapter the main concepts and technologies used in modern computer networks will be presented, with special emphasis on those used in the network of the association.

2.1 Definition of a computer network

A computer network is a packet switched telecommunication network composed by a set of hardware devices interconnected by communication channels. The task of a computer network is to deliver data exchanged by connected hosts: the information is transmitted as packets which are composed by an header, used by forwarding protocols, and a body which is the real information to be delivered to end users. The routing and forwarding process of packets is regulated by a set of protocols operating at different logical levels, which are going to be described in details in section 2.2

2.2 Computer network classes

Computer networks are classified according to their size and service type. Depending on the geographical extension, networks can be classified in 5 categories:

- **LAN - Local area network:** A LAN is a computer network that is limited to a single building. Usually its radius is on the order of 100 meters
- **CAN - Campus area network:** A CAN is a computer network that includes multiple LANs from different building of the same organization.

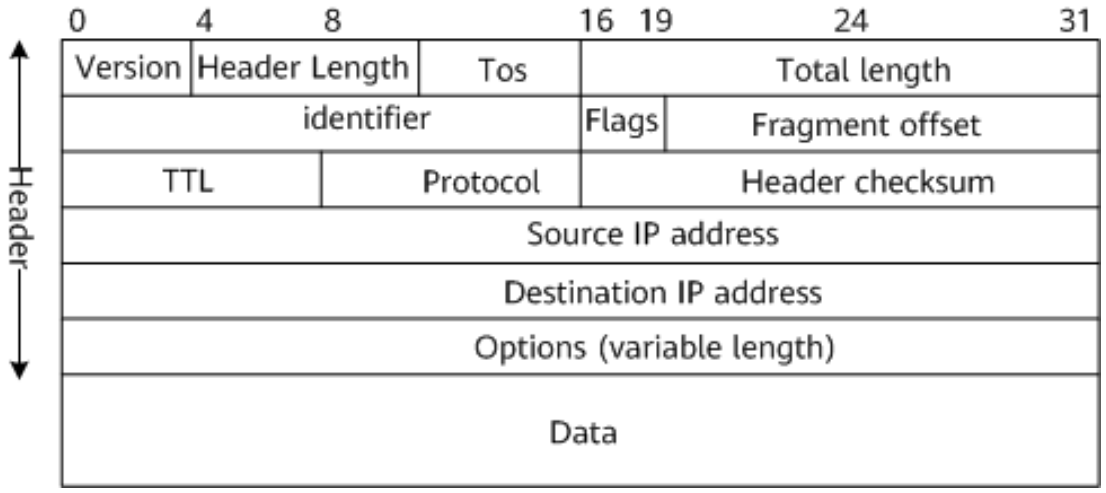


Figure 2.1: IP packet structure

- **MAN - Metropolitan area network:** A MAN is a computer network that spans across an entire city
- **WAN - Wide area network:** A WAN is a computer network that covers a large geographical area which include multiple cities and regions
- **GAN - Global area network:** A GAN is a computer network similar to a WAN, but with global coverage

Based on the type of offered service and connected hosts, a computer network can be further classified into 2 logical categories:

- **Local network:** A network that is operated and managed by a single organization. Examples of local networks are home networks, office networks and campus networks. Usually the size of this kind of networks is small as they are classified as LANs
- **Public networks:** A network that is owned and operated by an ISP in order to offer telecommunication services to residential and business customers. It is composed of a *transport network*, also known as *backbone network*, which is used to carry big amount of traffic between the network centers of the ISP, and of an access network, which is used to connect end customers to the network

2.3 The ISO-OSI model

The ISO/OSI model [2] is an open standard defined by the International Organization for Standardization with the aim of defining a layered logical architecture for

computer networks. The standard defines 7 layers: each one depends on network functions from the lower one and provides services to the upper one. Overall, the layers provide all the functions that are required in a computer network to exchange information between 2 interconnected hosts of different vendors.

- **Layer 7 - Application:** The application layer is the interface used by the user to interact with the system
- **Layer 6 - Presentation:** The presentation layer performs data formatting and translation when exchanging data between the application and the lower network stack
- **Layer 5 - Session:** The session layer manages, setup and terminates the session between two network hosts
- **Layer 4 - Transport:** The transport layer provides point to point connection between two directly connected hosts
- **Layer 3 - Network:** The network layer provides functionality to transmit information between nodes in different networks, thus not directly connected
- **Layer 2 - Data link:** The data link layer provides point to point data transfer between two directly connected nodes. It may also detect and correct possible errors
- **Layer 1 - Physical:** The physical layer is in charge of transmitting and receiving a raw stream of bits over a physical channel. It converts digital bits into physical signals

When moving down the network stack, the information is encapsulated in a packet of the corresponding layer by adding a new header which is stripped from the packet when moving up. Network nodes typically implements only functionalities from the first three layers while end hosts operate in all layers.

2.4 Network devices

A computer network is usually composed of a set of network devices connected by means of communication channels of different kind. Each class of network devices performs a specific elaboration on the flow of packets or bits. The following list describes the most common device classes:

- **Hubs:** A network hub is a device operating in the first layer of the ISO/OSI stack used to connect different physical communication channels. Since the

data received from a channel is forwarded to all the channels connected to the same hub, the collision domain of the network is the same. In modern networks, they have been almost entirely replaced by switches.

- **Switch:** A switch is a network device operating in the data link layer used to connect different physical communication channels while reducing the size of the collision domain. They operate at packet level by forwarding packets only to the intended hosts based on the information stored in the packet header and in the forwarding database of the switch itself. A switch can be *managed* or *unmanaged*, depending on the presence of a user interface.
- **Router:** A router is a network device operating in the network layer and it is responsible of forwarding IP packets between different IP networks. It uses the addresses stored in the IP header and in the routing table to forward packets towards the correct destination. The logical architecture of a router is usually divided in two parts, a *control plane* and a *data plane*: the first one includes the operating system, the routing protocols and all the software that is in charge of populating the routing table, which is then used by the second one to determine the outgoing interface for a given packet that reaches the router. Since the data plane performs always the same kind of operations while forwarding packets at very high speeds, it is usually implemented in hardware through application specific integrated circuit (ASIC) while the control plane is implemented in software and executed by a general purpose CPU (Central processing unit), enabling the possibility to upgrade the software and to add new functionalities to the device. On lower end devices, in order to reduce costs, all the forwarding operations can be performed by the router CPU (Central processing unit).
- **Firewall:** a firewall is an hardware or software component that is in charge of protecting a network segment by blocking all the traffic that matches a set of rules defined by the operator. Very simple firewalls, like stateless packet filters, operate in the lower layers of the network stack performing simple operations at very high speeds. By moving up the stack, the complexity of the operations that can be performed increases at the expense of elaboration speed. A firewall can also be used to perform network operations like network address translation (NAT) between two IP networks. Firewalls can be standalone devices but usually they are integrated in routers

2.5 Layer 2 networks

A layer 2 network is composed by a set of nodes directly connected by means of a single physical link.

Network switches that operate in the data link layer run a series of algorithms and protocols needed to populate the forwarding table and to forward packets from the source port to the correct destination port. The most important ones are the *backward learning* and the *spanning tree*

2.6 Layer 2 protocols

2.6.1 Backward learning

The backward learning protocol is an adaptive protocol used by network switches to build a forwarding table by simply analyzing the addresses contained in the packet headers. Little to no configuration is required by the network operator to make the protocol work and, because of that, most network switches available on the market are unmanaged. The main principle behind the algorithm is that if a packet with source address A is received from port B, then it is safe to assume that destination A can be optimally reached through port B. When a packet reaches a switch, two operations are performed by the device:

1. If the source MAC (Medium Access Control) address is unknown to the switch, a new entry is created in the forwarding table containing the source MAC address, the source port and a timeout set at *maxAge*. Otherwise, if the entry is already present in the table, the timeout is reset at *maxAge* and, if different, the source port is updated

2. If an entry with the destination address is present in the forwarding table, the packet is forwarded through the port stored in the table, otherwise it is flooded to all ports except the source one. If the destination port is the same as the source one, the packet is simply dropped. This ensures that, eventually, the packet will reach the intended destination because it is going to be transmitted in every network segment

By means of the packet flooding and source port updating techniques, the protocol also supports host mobility and network topology changes provided that, after a topology change, a packet from involved hosts reaches every switch in the network. The main problem of the protocol is that it works as intended only in networks with a tree-like topology: in presence of loops, a broadcast storm is triggered when a packet is flooded on all the ports of a switch and the network becomes unusable. To solve this problem, the spanning tree protocol has been introduced.

2.6.2 Spanning tree protocol

The spanning tree protocol has been introduced with the purpose of transforming each network topology in a tree-like topology, thus eliminating all the problems

caused by broadcast storms. After a transient period following the switching on of the switches in the network, the protocol disables all the ports that are part of a loop.

2.6.3 VLANs

Another problem of layer 2 networks is the extension of the broadcast domain. In small office / home office (SO-HO) environments this is not an issue because the number of devices is usually limited and it is rare to find networks with more than 30-40 network devices. In large networks (CAN, WAN, MAN) the number of devices in the same broadcast domain can be really large: for example, in the network of SFSC there are more than 2000 network devices and about 8200 CPEs: in such large networks, the broadcast traffic may be relevant, wasting computational resources in network devices and bandwidth. It is not always possible to limit the size of networks because of topology or administrative constraints, but broadcast domains can be logically separated by using VLANs. Multiple VLANs can be created in a single physical network, thus allowing to have multiple logical local networks and broadcast domains. In order to enable multiple VLANs in the same link, frames must be *tagged*: an additional header, standardized by IEEE 802.1q, is added to the MAC header which contains, among other fields, the VLAN ID. In switches, VLANs are implemented by adding another column to the forwarding database which contains the ID associated to the packet from which the source address was learned: when forwarding a tagged packet, only entries tagged with the same ID are considered for lookup. Nowadays, VLANs are supported in almost all enterprise switches

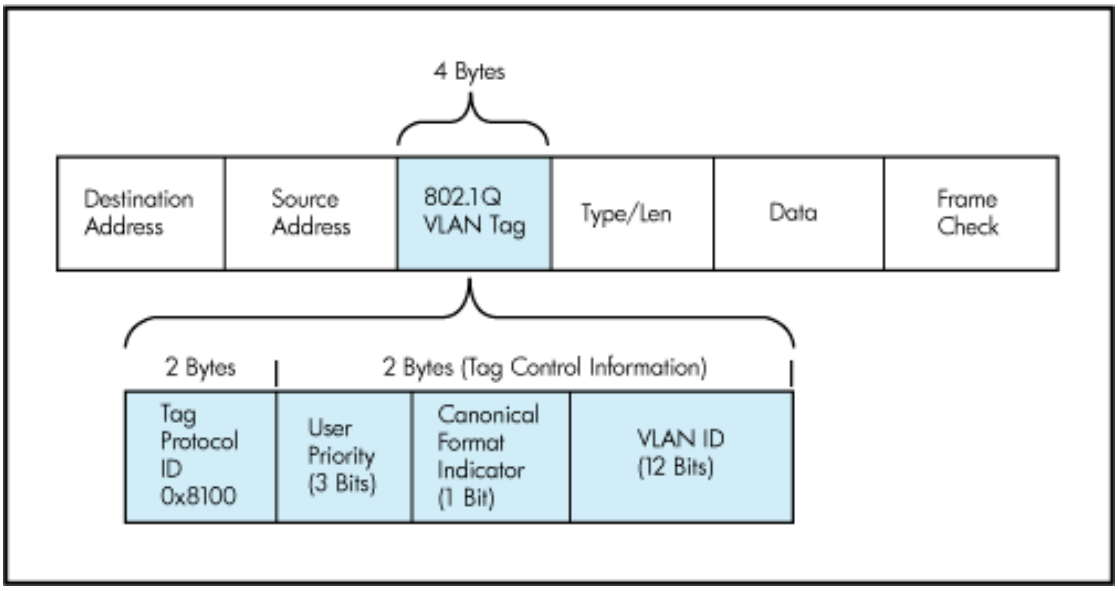


Figure 2.2: VLAN-tagged ethernet frame

There are multiple methods to associate traffic to a specific VLAN:

- **Port-based assignment:** All the traffic received from a specific port belongs to the same VLAN
- **Transparent assignment:** Assign traffic based on different criteria (source MAC address, L3 protocol)
- **Cooperative assignment:** Users set VLANs on their NICs

Links can be of different types:

- **Access:** Access links receive and transmit untagged frames and are usually used to connect end hosts. Each access port has one and only one associated VLAN: frames are tagged when received and untagged when transmitted.
- **Trunk:** Trunk links transmit only tagged frames and usually are used to connect switches. A trunk port can be configured both to allow the forwarding of all VLANs or to filter some of them.
- **Hybrid:** Hybrid links carry both tagged and untagged traffic. Untagged traffic is assigned to the native VLAN when received by a switch, which is always stripped when the packet is retransmitted on another trunk link. Hybrid ports are used to connect VLAN-aware switches to non-VLAN-aware ones.

Since tagged frames cannot cross VLANs, in order to let two hosts belonging to different VLANs to communicate, a router must be used and different IP networks must be configured on different VLANs.
Another possible use case for VLANs is to optimize network paths: by filtering specific tagged frames on trunk ports, network traffic can be forced to take a path instead of another, thus enabling the transit of different IP networks on different physical links.

2.7 Layer 3 networks

When the network size grows, it becomes difficult to manage a single L2 network and many problems arise:

- All the traffic is forwarded according to the same spanning tree and the usage of network resources is not fair since all the links that are outside of the spanning tree are not used
- Possible broadcast storms
- Limited network diameter because of spanning tree protocol limits (no more than 7 cascading switches)
- No network isolation
- Flat addressing: every switch must know the MAC address of each host since they can not be summarized

A possible solution is to split a large L2 network into smaller ones, each one associated to a different IP network, with routers as gateways to route traffic between the networks.
Conceptually, a router works similarly to a switch, but with some key differences:

- Packet forwarding is performed according to IP addresses and routing tables
- The backward learning protocol is replaced with static routing or dynamic routing protocols, hence a router is a managed device which needs to be configured in order to work
- The routing table contains a set of prefixes (IP networks) associated with the address of the next hop router or the name of an interface. Matching is done according to the longest prefix match rule
- Since a router is a complex device, the routing process can be fine-tuned by the network operator

The main drawbacks of L3 networks are the increased complexity of packet processing, which makes the devices more expensive, and the loss of a plug-and-play behaviour, typical of L2 networks, but L3 networks come with a set of strength points:

- Networks isolation
- Addressing plan defined by the administrator
- Unlimited network diameter thanks to hierarchical addressing
- Packet routing is not limited to a single forwarding tree and multipath routing is available, hence the same destination can be reached from multiple paths, increasing fault tolerance and bandwidth availability

2.8 Layer 3 forwarding and routing protocols

The following are the main technologies, routing and forwarding protocols used in a IP network:

2.8.1 Longest prefix match and hierarchical routing

An IP address is composed by two parts: an host part and a network part, also known as prefix. Prefixes identify IP networks in a hierarchical way: the same network can be included into more prefixes with different lengths, with the longest one being more specific than the others. When a packet reaches a router through an interface, a routing decision has to be taken: the router strips the host part of the destination address contained in the IP header and compares the prefix with the ones stored in the routing table. Given that a network can be identified by multiple prefixes, the router have to choose a single output port between the possible ones and this choice is performed by preferring the longest prefix that matches the destination IP network.

2.8.2 Routing protocols

Since routers are managed devices, routing tables are populated either manually by the operator or in a dynamic way by routing protocols

Static routing

It is the simplest way to populate the routing table: routes between nodes are defined manually by the network operator. An IP network configured with static

routing is not fault tolerant because the network can not react to topology changes or outages, therefore requiring manual intervention by the network administrator to work again. Static routing is suitable only for small and simple networks, where the usage of routing protocols would be excessive.

Dynamic routing protocols

In large IP networks with complex topologies, where it is important to have a dynamic behaviour with a semi automatic configuration of devices and fault tolerance, dynamic routing protocols are used. By exchanging information about reachable destinations with neighbor devices, a dynamic routing protocol builds the network map and then installs the routes in the routing table based on specific selection criteria. Being dynamic, it is also able to react to topology changes or faults without the intervention of the administrator. There are two types of routing protocols:

- **Link state protocols:** Link state protocols send information to all the routers in the network about the directly connected networks, allowing each router to have a complete picture of the entire network topology. After having built the network graph, the routing protocol computes the shortest path to each destination and installs it in the routing table. Different metrics can be used as *distance*: number of hops, bandwidth, delays and user defined weights are the most common ones. Another key feature of link state protocols is the automatic neighbor discovery.
- **Distance vector protocols:** Distance vector protocols send their routing table only to directly connected neighbors: by combining all the information learned from neighbors and the router's own routes, the optimal route to a given destination is installed into the routing table. Distance vector protocols have a very detailed knowledge about nearby network which decreases as the distance increases because of the IP networks summarization.

Two of the most common protocols in modern networks are Open shortest path first (OSPF) and Border gateway protocol (BGP)

OSPF

[3] The OSPF protocol is a link state dynamic routing protocol used to exchange routing information between hosts belonging to the same Autonomous system (AS). Since it is a link state protocol, it floods the information about its directly connected networks to all the relevant routers using LSAs (Link state advertisements) and it uses the Dijkstra algorithm to find the shortest path to a given destination. One of the key features of OSPF is the hierarchical partitioning of the network in

multiple areas in order to support big and complex networks. Each router has a full knowledge of the network inside its area but only summary information from networks in other areas, thus reducing the number of flooded LSAs in big networks and resources usage in routers. In order to find the best path, OSPF uses link costs, a metric defined by the network administrator. In case of unit costs for all the links, the protocol chooses as shortest paths the ones with the least number of hops. OSPF defines different types of areas:

- **Backbone area:** It is the core area of an OSPF network. All other areas are connected to the backbone area, which is where the intra-area routing occurs
- **Standard area:** It is an area in which LSA packets can be normally transmitted, even if coming from different areas
- **Stub area:** It is a kind of area that does not accept any route learned from different protocols, like BGP routes or static routes. To reach those destinations, a default route must be used
- **Totally stub area:** Similar to the stub area, but it does not accept OSPF routes from different areas
- **Not so stubby area:** A stub area that is able to export OSPF routes to the backbone and to other areas

Different classes of routers exist in an OSPF network according to their adjacencies:

- **Area border router:** A router that links multiple areas. I keeps a copy of the database of each area
- **Backbone router:** A router with at least one interface towards the backbone area
- **Autonomous system boundary router:** A router connected to other autonomous system
- **Internal router:** A router with adjacencies to other routers of the same area
- **Designated router:** In a L2 broadcast domain, it is the router which is in charge of flooding OSPF messages to other routers

The algorithm by itself is quite complicated, this is an high level look of the various steps:

1. Upon initialization or on topology changes, a router generates an LSA which contains all the link-states of the router

2. All routers exchange their link-states and each one builds the database based on received LSAs: since they are flooded, the links database is the same in all routers

3. After the route database in each router is completed, the router calculates the shortest path tree to all the destination by placing itself as root and using the Dijkstra algorithm

OSPF also support multi-path routing in case of equal costs to a destination

BGP

[4] BGP (Border gateway protocol) is an EGP (Exterior gateway protocol) used to exchange routing information between routers belonging to different AS and administrative domains. Even if it is normally considered as an EGP, BGP could also be used to simplify the internal networks interactions of large companies or ISPs thanks to its large list of configuration parameters. With specific configurations, it could also be used to perform basic traffic engineering. BGP sessions, unlike what happens in OSPF, must be configured manually by an operator. Two types of sessions exists:

- **eBGP:** A session established between routers of different AS, used to learn prefixes of the other organization
- **iBGP:** A session established between routers of the same AS, used to redistribute routes learned from the other AS to the core routers

BGP works by announcing to neighbors a set of prefixes reachable by the router. Unlike link-state protocols, the router does not announce its full routing table: route announcements can be manipulated by network administrators based on technical or administrative policies. Together with prefixes and their length, a set of metadata are included in BGP announcements to perform policy-based routing. A BGP router may receive multiple announcements towards the same destination from different AS, but at most one path will be selected by executing the *BGP selection algorithm*, which considers different policies configurable by the operator. The following is an ordered list of the attributes evaluated by the BGP selection algorithm to determine the best next hop:

1. Local preference

2. Locally originated networks or aggregated networks are preferred

3. AS path length

4. Lowest origin type (IGP, EGP, Incomplete)

5. Lowest multi-exit discriminator
6. Prefer eBGP over iBGP
7. Lowest IGP metric
8. Prefer the oldest route

MPLS (Multiprotocol label switching)

[5] When a packet traverse a conventional IP network, each router performs a series of operations on the IP header in order to define the exit port, with the main one being the longest prefix match. By using the MPLS technology, a short label is assigned to each packet when it enters a MPLS core network and the forwarding process is performed based on that label, without the need to examine the IP header before each hop, thus improving performance and throughput of the forwarding process. When an IP packet enters a MPLS network, the LSP (label switched path) must be valid and established, with all the involved routers agreeing on the hop-by-hop sequence of labels, since they are replaced at each hop: to reach this goal, the LDP (Label distribution protocol) is used. A variant of MPLS, named MPLS-TE (Traffic engineering), allows for more granular control over traffic flows, enabling network administrators to optimize traffic paths and to guarantee a certain amount of bandwidth to each LSP. It performs load balancing across multiple paths and it also allows the preallocation of backup links for fast rerouting in case of link failures. Given the high scalability and efficiency of MPLS, together with its large amount of features for traffic engineering, it is often used in large core networks to increase efficiency and manageability.

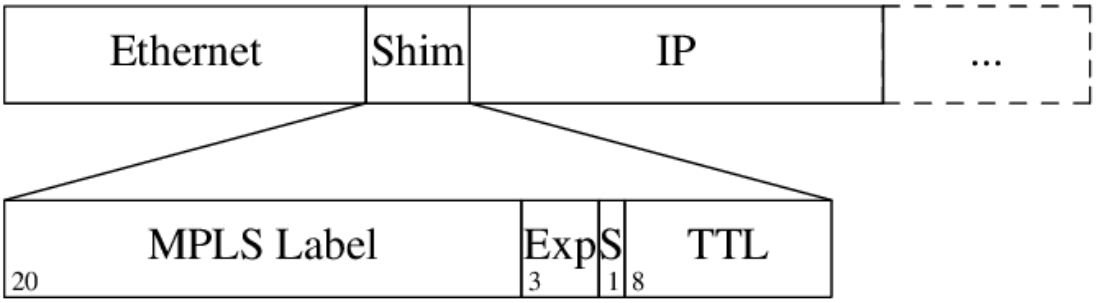


Figure 2.3: MPLS header structure

Chapter 3

SFSC network state of the art

This chapter contains an overview of the network topology and configuration before implementing the results obtained in this thesis project.

3.1 Network topology

The network of SFSC is a WAN with the goal of providing fixed internet wireless access to residential customers. The network is made of about 300 nodes and 2000 devices, usually hosted in bell towers, private houses and buildings and it's constantly growing. Each node is connected with many downstream and upstream nodes by using radio links operating in unlicensed frequency bands. An essential requirement is the direct visibility between nodes because of the used frequencies and transmitting power. This constraint, coupled with the complex orography of the territory in which the association operates, poses a series of limitations about the link distance, nodes position and technology that can be used. The network is composed of two logical parts:

- **Access network:** The access network is composed by all the nodes directly connected to end users

- **Transit network:** The transit network comprehends all the nodes used to provide connectivity to access nodes and to other transits: it is the core of the network. Transit nodes are connected in a partially meshed topology with high capacity, point-to-point radio links.

Figure 3.1: a) An hybrid node in Villadeati (AL), b) A CPE on the rooftop

Different kind of network nodes exist. Each one is composed of a set of radio and switches connected by means of ethernet cables. Most of the radios are powered using the PoE (Power over ethernet) technology in order to reduce the number of cables that must be laid in each installation.

- **Access nodes:** This kind of node is part of the access network. Typically an access node hosts one or more upstream wireless links towards the transport network and a variable number of access radios used to connect end users in a PTmP scheme. The number of access radios depends on the number of connected users: when an access point is created, only a small number of access radios is installed with more being added if the number of connected users increases. On average, a single radio provides connectivity to 10-15 users up to 40 in case of crowded sites. Associates connect to the access network via a CPE (Customer premise equipment) which is a radio with an integrated router installed on the rooftop of the house or in any place with direct visibility with the access points. The access link maximum distance is of two kilometers up to four in case of visibility problems.

- **Transit Nodes:** A transit node is part of the transit network and its purpose is to relay bandwidth to other transit or access nodes. Typically in a transit node we can find multiple high capacity radio links and multiple switches or routers. Transit nodes may also host access radios.

- **Fiber optic switching point:** A fiber optic switching node is a site in which the wireless network connects to the internet through a fiber optic cable. It contains an high performance router which acts as a gateway for the network section that depends on it and a large number of high capacity bridges. Currently there are four fiber optic switching points in the network (in the cities of Ivrea, Casale Monferrato, Asti, Saluggia) and each one handles the traffic of a single network area which is logically independent from the

others. The four networks meet in a limited number of transit nodes but the traffic is completely isolated since they are on different L2 domains.

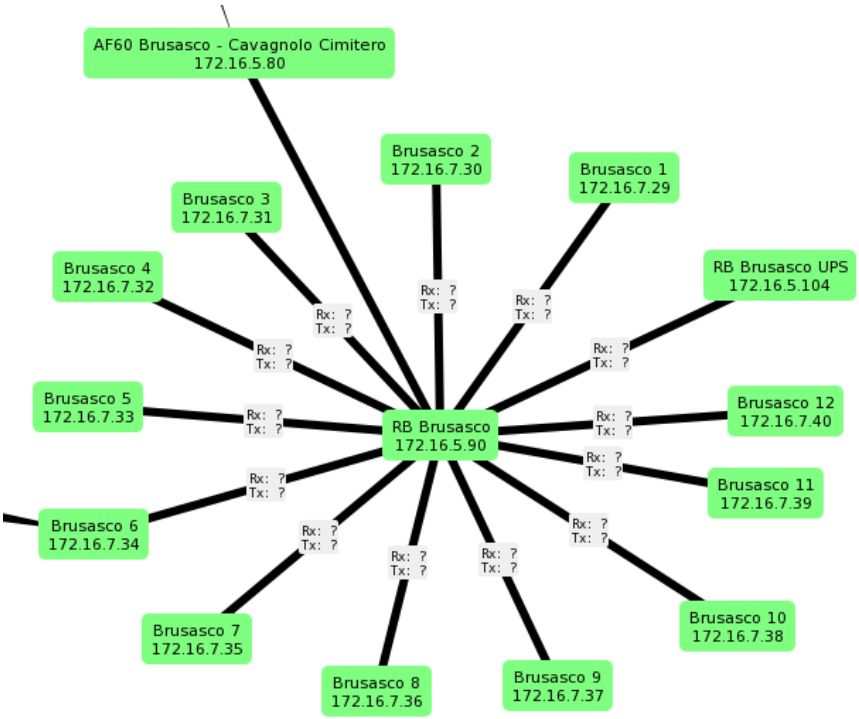


Figure 3.2: One of the largest access point in the network. In the center we can see the access point router, surrounded by access radios. This picture is just a scheme and does not represents the real position of the devices in the tower

Network nodes are organized in a tree topology with a very limited number of loops

Nodes layout

Depending on the size of the node, different layouts are used:

- **Single router:** This layout is used in small access/hybrid nodes with few devices carrying a low amount of traffic. The router is used to connect both access radios and transit bridges, if any.
- **Switches and routers:** This layout is used in hybrid/access nodes with a small number of devices. Usually the switches are used to connect transport bridges while the routers takes care of handling the access network traffic.
- **Big switch:** This layout is used in nodes where a large number of devices

must be connected. A single switch with many ports is used to connect both bridges and access radios.

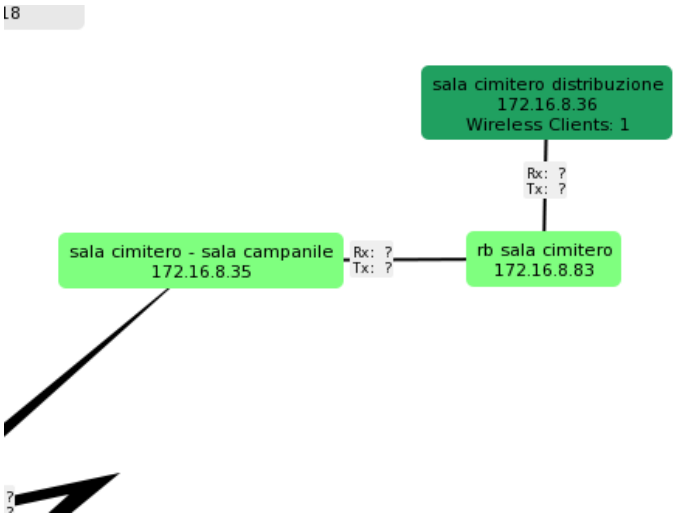


Figure 3.3: Single router: 172.16.8.32 is the upstream bridge, 172.16.8.83 is the router and 172.16.8.36 is the access radio

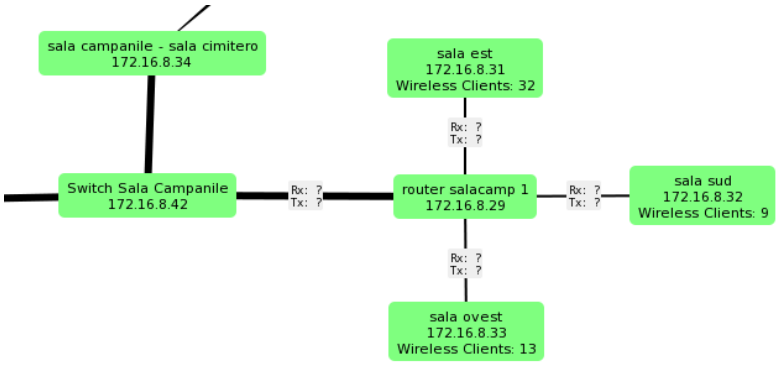


Figure 3.4: Switch plus router: 172.16.8.29 is the router, 172.16.8.42 is the switch, 172.16.8.34 is a downstream bridge and the remaining devices are the access radios

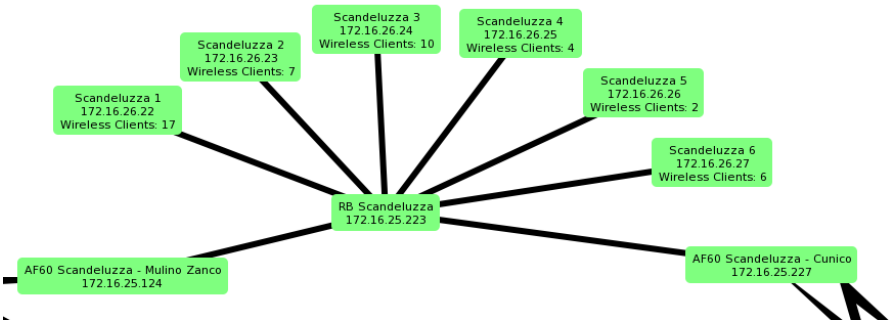


Figure 3.5: Big switch: 172.16.25.223 is the switch while the other devices are access radios and bridges

3.2 Wireless devices

3.2.1 Frequency bands

The network of SFSC is an entirely wireless network, both in the access and transport segment. Different types of radio devices are used, mainly depending on the cost, link distance and bandwidth requirements. All the devices operate in this unlicensed frequency bands:

- **5 GHZ:** This band is used by low to medium capacity transport links and access radios. It provides excellent stability in rainy conditions and long distance links, but nowadays the 5 GHz spectrum is crowded and unreliable in densely populated urban area because of interference. Both 20 MHz and 40 MHz wide channels are used, depending on the required capacity and RF noise in the area.
- **24 GHz:** This band is only used for medium capacity, long distance transit links. It is very resistant to interference, but it is unreliable when used in long distance links during bad weather conditions because of the attenuation caused by water and vapor in the air
- **60 GHz:** This band is mainly used by high capacity, medium distance transit links and a few experimental access radios. It is almost immune to interference, but the performance of links operating in this frequency band quickly degrades as the distance increases and in case of rain. 60 GHz medium to long distance transit links are usually coupled with an auxiliary 24 GHz or 5 GHz bridge which act as a backup in case of failure of the main one due to rain.

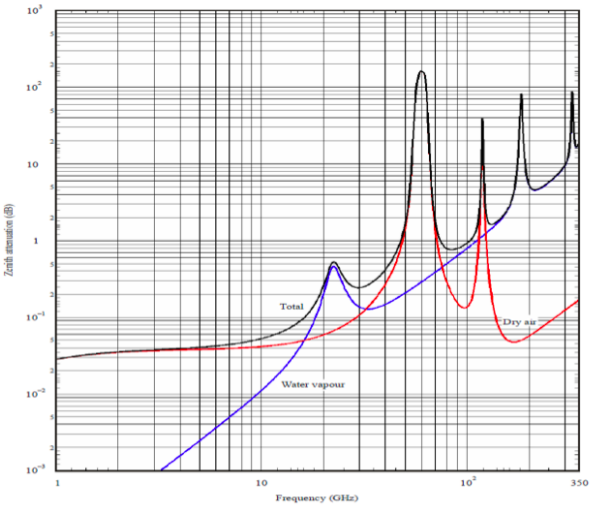


Figure 3.6: Attenuation of radio signals in the athmosphere. Note the two absorption peaks in the 24 GHz and 60 GHz bands [6]

3.2.2 Radio types

Different kind of radios are used in the network, all of them manufactured by Ubiquiti and Mikrotik. Ubiquiti devices are mainly used in P2P and access links while Mikrotik ones are used in access links, switches and routers. The following is a list of the main devices used in the access network:

- **Mikrotik RB922UAGS-5HPacD:** An access radio operating in the 5 GHz band, running RouterOS. It supports the IEEE 802.11 a/b/g/n/ac standards and it uses a proprietary TDMA (Time division multiple access) protocol in order to solve the "hidden node" problem and increase overall throughput. It is usually coupled with a symmetrical horn antenna characterized by a great side lobes attenuation which allows for better frequency reuse and noise immunity.[1]

- **Mikrotik BaseBox 5:** An access radio similar to the Mikrotik RB922 without the IEEE 802.11ac standard support. This device is obsolete and it is currently being replaced by the RB922 radio.

- **Ubiquiti Rocket Prism 5AC:** An access radio operating in the 5 GHz band

[1]The hidden node problem is caused by a lack of visibility between nodes in the same wireless network which causes collisions when transmitting over a shared channel

that shares some characteristics with the Mikrotik RB922 like the TDMA technique and the 802.11ac standard.

Ubiquiti devices are used in new installations because they are more resilient to noise and they behave better with a large number of connected clients. As CPEs, users receive a device which connects to the access radio and behaves like an usual home gateway. It performs NAT on the LAN side and handles the assignment of IP address to the devices connected to the ethernet interface through a DHCP server. Since the TDMA protocols used by the two vendors are incompatible, CPEs from the same vendor of the access radio must be used. In the transport network, the following are the main devices used to realize P2P bridges:

- **Ubiquiti PowerBeam 5AC:** An half duplex radio with integrated parabolic antenna used in medium range links (less than 6-7 kilometers). It operates in the 5 GHz band and it has a maximum aggregated capacity of about 200 Mbit/s when using a 40 MHz wide channel.
- **Ubiquiti AirFiber 5XHD:** A bridge used in medium/long range links operating in the 5 GHz band. It has a maximum aggregated capacity of 350 Mbit/s when using 40 MHz channels and it comes with a set of advanced features:
  - Frequency split for uplink and downlink
  - Possibility to modify the downlink to uplink ratio, thus supporting asymmetric traffic flows
  - Advanced RF filtering
  - **Ubiquiti LTU LR:** A low power version of the AirFiber 5XHD designed to be used as a CPE with P2P support normally used in short or medium distance bridges.
- **Ubiquiti AirFiber 24**: A full duplex radio used in medium/long range links. Since it operates in the 24 GHz band, it is almost immune to interference but it suffers from attenuation during rain or thick fog conditions in long-distance links. It has a maximum capacity of about 700 Mbit/s up to a distance of 10 kilometers. This device only supports two channels, one for the downlink and one for the uplink when operating in full duplex mode, therefore great care must be taken when devices are co-located in the same tower in order to avoid interference.
- **Ubiquiti AirFiber 60 LR**: An half duplex radio used in short/medium range links, operating in the 60 GHz band. It suffers from great attenuation by oxygen and water vapour in the atmosphere but it can deliver up to 2

Gbit/s of aggregated traffic. It is the best choice for short/medium range, high throughput links (less than 4 kilometers) and it is usually coupled with a 5 GHz or 24 GHz bridge in case of failures due to bad weather conditions. Since in Italy the device is only allowed to operate in the 57-64 GHz band, and the oxygen attenuation is maximum at 60 GHz, usually only the highest channel is usable: as for the AirFiber 24 bridges, they must be carefully installed when co-located.

3.3 Network devices

All the wireless radios are interconnected by means of routers and switches. All the network devices are manufactured by Mikrotik and most of them are equipped with RouterOS, an operating system based on Linux.

3.3.1 Switches

The following is a list of L2 switches in use in the network.

- **Mikrotik RB260GSP:** The most common L2 switch in the network. It does not have L3 capabilities except for management purposes (performed via an HTTP dashboard), therefore it only understands L2 frames. It has 5 gigabit ethernet ports, 4 of them with PoE-out support. This is a VLAN aware switch: trunk and access ports can be configured and VLAN filtering on specific ports is supported as well. It is mostly used in small nodes to connect wireless bridges and routers. The operating system is SwitchOS.
- **Mikrotik CRS112-8P-4S:** An 8 port switch with PoE support and limited L3 capabilities. It is equipped with RouterOS.

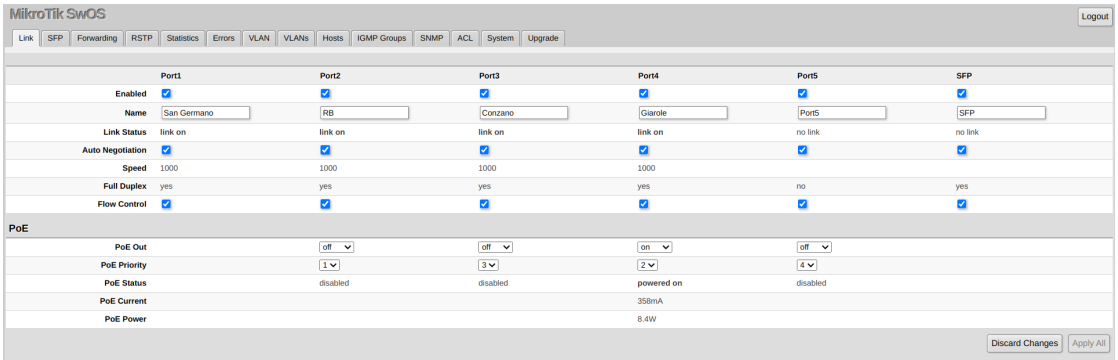


Figure 3.7: One of the SwitchOS management interfaces

3.3.2 Routers

This class of devices integrate both L2 and L3 capabilities. Interfaces may be configured in different ways:

- **Routed interface:** An interface which is not assigned to any bridge and has an IP address configured.
- **Switched interface:** When an interface is assigned to a bridge, it becomes switched and is part of the same L2 domain of the other bridged interfaces. They loose L3 capabilities which are transfered to the virtual bridge interface.

Each router has a variable number of interfaces that can be configured independently. A general purpose CPU is installed on each device and, in most of them, a dedicated switching chip is available to perform L2 operations. In general, routers are complex devices and, since they can also perform L2 operations, its usage is usually preferred over pure switches.

RouterOS

[7] RouterOS is the operating system preinstalled in all Mikrotik routers but it can also be installed on third party compatible hardware. Since it is based on Linux (version 5.6.3 starting from RouterOS 7) many of its functionalities follow those already implemented in the underlying kernel. The following is a list of the main features:

- **Bridge:** A virtual bridge can be configured and physical interfaces can be assigned to it, becoming slaves. A bridge represents a single L2 domain and all the features of traditional bridges can be configured on it, like VLANs, STP, backward learning protocol and others. From the point of view of the operating system, a bridge represents an L3 interface connected to a switch with all the bridged interfaces attached: IP addresses can be configured on it and it has full L3 capabilities. While in physical bridges all the forwarding operations are performed in hardware, RouterOS performs this operations using the general purpose CPU: on some specific router models, switching operations can be offloaded to the integrated switching chip, if available. RouterOS bridges fully support VLANs and all the related operations:
  - Independent VLAN hosts learning
  - Trunk port/access port configuration
  - VLAN filtering on specific ports
  - Virtual VLAN interfaces

- **Routing:** RouterOS supports static routing and a multitude of dynamic routing protocols like OSPF, BGP and RIP. Both IPv4 and IPv6 are supported. VRF (Virtual routing and forwarding) is also supported, allowing the coexistence of multiple routing tables in the same device: traffic can be marked based on different rules and forwarded according to the rules contained in different routing tables.
- **Routing filters:** Routing filters are used by routing protocols to manipulate incoming and outgoing network announcements. Routes can be accepted, rejected and the routing protocols attributes can be manipulated based on a set of properties accessible in the filter condition clause. The following is the structure of a routing filter:

```
1    if(condition is true) {action} else {action}
```

  Routing filters can be chained like firewall rules. The evaluation of the chain rules is stopped when the *accept* or *reject* action is met.

- **Firewall:** RouterOS firewall is based on the functionalities provided by the netfilter module implemented in the Linux kernel. It supports stateless and stateful packet filtering, application layer filtering, network address translation, packet and connection marking, packet mangling and raw packet elaboration. Connection tracking of different L4 protocols is also supported, allowing to perform operations on packets which require to keep track of the state of the connection, like NAT, stateful packet filtering or packet marking based on connections. The firewall is composed by a set of rules grouped in chains, which in turn are grouped into tables. Each rule is made of two parts: the first one specifies a set of conditions that the matching packet must satisfy, while the second one defines the action to perform on matching packets.
  Rules of the same chain are evaluated in order from top to bottom, until the end of the chain is reached or an action stops the evaluation of rules. The RouterOS firewall supports the following tables, each one defining the type of operations that can be performed on packets:
  - **Filter:** Used to perform filtering operations on packets
  - **NAT:** It contains the rules used to perform network address translation operations
  - **Mangle:** A table which contains the rules used to manipulate the fields of the IP header
  - **Raw:** Rules in this table are evaluated before the connection tracking is performed

Users have the possibility to define custom chains, but the following are already implemented by default:

- **Prerouting:** This chain is checked against all the packets entering the router from any interface. The prerouting chain is used to perform destination network address translation (DNAT) or stateless packet filtering in the raw table
- **Input:** This chain is matched against all the packets with a destination address same as one of those configured on the router.
- **Forward:** Packets traverse this chain when they are directed to remote destinations. When a packet is checked against this rule, a routing decision has already been taken.
- **Output:** This chain is checked against all the packets originated from the router
- **Postrouting:** This chain in checked after the routing decision has been taken. In is mainly used to perform source address translation (SNAT)

IPv4 and IPv6 protocols use different instances of the firewall. Another important feature of the firewall is the possibility of grouping IP addresses or prefixes in lists, which can be referenced in different sections of the router configuration. Lists can be populated either manually by the administrator or dynamically by specific firewall rules

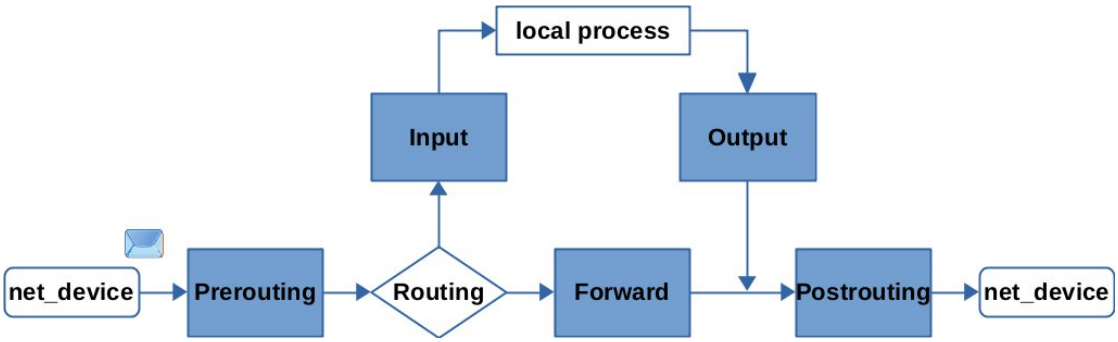


Figure 3.8: The flow of packets through the chains of the firewall [8]

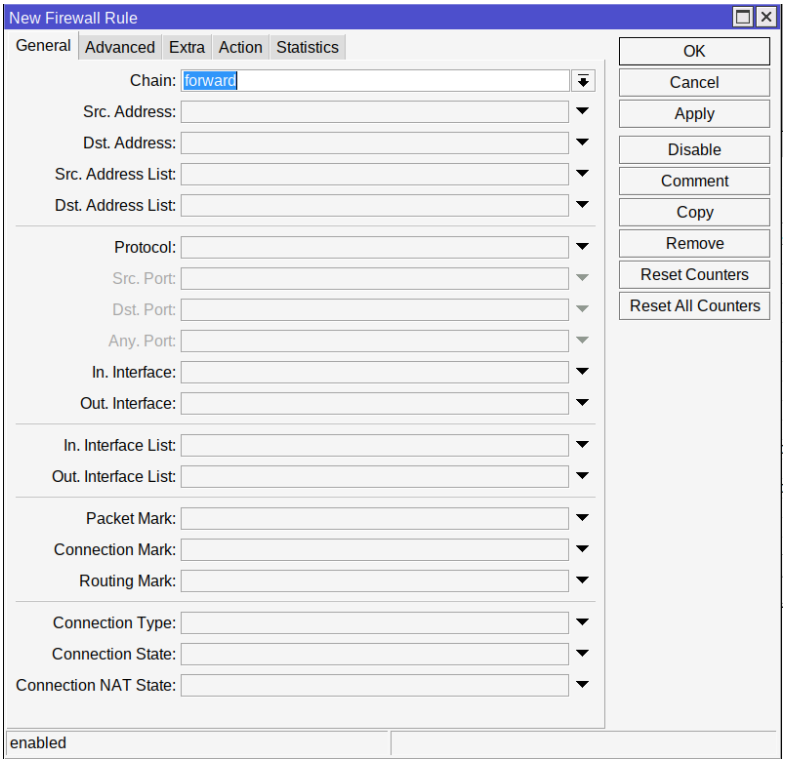


Figure 3.9: Some of the matching criteria available in the forward chain

- **VPN:** RouterOS supports the creation of VPN (Virtual private network) with different protocols like OpenVPN, Wireguard, L2TP/IPsec and others. In the network of SFSC they are mainly used to reach network devices and CPEs from remote locations when performing maintenance or inspections.
- **Queueing and QoS (Quality of service)**: In order to prioritize traffic flows and to shape network traffic, a set of QoS features are implemented in RouterOS. When coupled with the packet marking utility of the firewall, it is possible to finely select and shape traffic flows based on different parameters like source/destination addresses, ports and protocols. QoS is also used to evenly share the available bandwidth between the users, especially in case of over subscription.
- **Scripting:** RouterOS supports the execution of scripts written in a proprietary, feature-rich scripting language. Scripting provides a way to automate some router maintenance tasks by means of executing user-defined scripts bounded to some event occurrence. They are particularly useful when performing bulk updates of configurations or to perform operations when a particular condition is satisfied or at constant time intervals, like cron jobs.

Different interfaces are available to interact with RouterOS:

- Proprietary Winbox management software which allows to directly connect to devices via IP or MAC address.
- SSH
- Telnet
- Telnet over MAC
- API for integration into existing software
- Web Interface

Router models

All the Mikrotik routers runs RouterOS versions between 7.6 and 7.11.2 with the same feature set. They mainly differs in performance, number and type of ports, and form factor. These are the main products used in the network:[9]

- **hEX PoE:** A low-end router with 5 Gigabit Ethernet ports with PoE support and 1 SFP port. Usually it is used to connect access radios and switches in small and medium size access points and it acts as a PPPoE concentrator by establishing sessions with the CPEs of the users. The hEX PoE is usually deployed in access nodes with less than 100 users.
- **RB5009UPr+S+IN:** A router equipped with 8 Ethernet ports with PoE support and an SFP+ port. Because of its modern quad-core ARM CPU running at 1,4 GHz, it is commonly used in large access nodes with a substantial amount of IP traffic being routed between the transport and access network
- **CRS318-16P-2S+OUT:** A 16 port router with PoE support and an advanced switching chip capable of performing hardware L3 routing, allowing this device to reach wire-speeds when forwarding IP packets. It also features a 700 MHz ARM CPU capable of providing decent performances when routing packets that can not be offloaded to the CPU. This device is mainly deployed in access nodes with a large number of radios or in transit nodes which handle a large amount of traffic
- **CCR1072-1G-8S+RM:** A powerful router used in the core network. It features 8 SFP+ port, each one with a capacity of up to 10 Gbit/s and a 72-core CPU, which is particularly efficient when performing bandwidth-shaping and NAT on a large number of flows

Except the CCR routers, all the others are equipped with a switching chip that takes care of all the L2 operations.

3.4 Network configuration

This section contains an overview of the network configuration as it was before the migration to the new one.

3.4.1 Network areas

In the network there are four fiber optic switching points, with each one being the root of a single, independent network area. The traffic of each network area is handled by the area border gateway which performs the following tasks:

- Firewall
- NAT of the users and network devices private IP addresses to the public IP pool
- Bandwidth shaping
- Gateway towards the internet for the access point routers
- Gateway for the management network of the area

3.4.2 Management network

All the devices need to be directly reachable by network administrators and monitoring systems in order to perform monitoring, troubleshooting and management. All the traffic generated by management tools is usually transmitted over a dedicated L2 network which can be of different types:

- **In band management:** The management network shares the same communication channels of the main one, but it is logically isolated by means of VLANs or other isolation techniques. This kind of network is easier to deploy because it exploits existing communication channels, but it can be interrupted in case of failures in the main one. In case of congested links because of high user traffic, some devices may be difficult to reach unless some QoS techniques are implemented and an higher priority is assigned to management traffic.
- **Out of band management:** The management network is physically isolated from the main one and it exploit different communication channels. This kind of network guarantees the highest availability, but it is more expensive and diffucult to maintain.

The management network of SFSC is an *in band* network, exploiting VLANs to isolate user and management traffic. Each one of the four network areas has a

separate management VLAN with different private IP subnets: all the network devices of the area are connected to the same L2 domain identified by the VLAN ID and a static IP address is manually assigned to each one.

VPNs and network monitoring system

A VPN (Virtual private network) infrastructure is used in order to allow network administrators and tools to reach network devices in different physical locations. The OpenVPN protocol is used because of its security and multi platform support

- A RouterOS cloud instance acts as a VPN server and network monitoring system by running The Dude application
- Network operators are connected to the VPN server in a road warrior configuration while border gateways use a site-to-site VPN to route traffic between the management networks and VPN users
- Static routes are configured in the VPN server to reach management networks through area border gateways

3.4.3 Transport network

The transport network is responsible of forwarding user traffic between the access point routers and the area border gateway. It is an L2 network and uses VLANs to create an isolated L2 domain between the area border gateway and the access point router. A small private IP subnet is configured on each transport VLAN (/29 or /30 depending on the number of access point routers) and static routes are configured:

- A default route to the area border gateway is configured in each access node router
- Routes to user networks are configured in the border gateway, with the corresponding access point router IP as next hop

As described in section 3.1 the network has a tree topology with a small number of loops: in order to avoid problems caused by broadcast storms, a pure tree topology must be enforced. The Spanning tree protocol may be a solution, but it presents some issues that discourage its usage:

- The STP generates a single forwarding tree by completely disabling ports that cause a loop, wasting resources in unused links
- Long convergence time

- The STP is a protocol designed for small LANs. In a WAN with a large number of hops, it is common to exceed the default limit of 7 hops so a manual configuration of the parameters is required in each bridge to avoid undefined behaviour.

To overcome this issues, the spanning tree is disabled in the entire network and the manual VLAN filtering technique on trunk ports is used. In order to break loops, the following rules are enforces in the switches:

- Since both management and user traffic are VLAN tagged, untagged traffic is filtered on all ports
- Loops in the management and transit VLANs are interrupted by filtering frames on specific ports
- In order to exploit all the available bandwidth, transit VLANs are filtered on different ports when multiple links are available. Bandwidth requirements of the access points are taken into consideration when distributing the traffic among different links in order to avoid traffic imbalance.

This solution still present some issues that will be discussed in details in chapter 5.

3.4.4 Access network

The access network spans between the access point routers and the user CPEs. It is an L2 network with VLANs used to isolate L2 domains between user CPEs connected to the same access radios and routers. In order to enable point to point IP traffic between the user CPE and the access point router, a PPPoE session is established on top of the VLANs. Authentication, authorization and accounting is performed via RADIUS: the client component is hosted in the access point router while the server is hosted on a remote Ubuntu server.

3.4.5 Addressing plan

All the devices in the network are configured with private IP addresses. Different addressing schemes are used:

Public IP addresses

SFSC owns the 185.168.96.0/22 network which is split in four /24, each one allocated to a different network area and announced to the upstream provider via eBGP.

Management networks

A subnet in the 172.16.0.0/16 address range is assigned to each management VLAN. The size of the subnet depends on the number of devices that need to be addressed, ranging from /23 in a portion of the Saluggia area to /21 in the Casale area. All network devices are configured with an IP address taken from the management network of its area. Since network devices need to access the public internet (to fetch updates, clock synchronization, DNS and other maintenance tasks) each management network is mapped to a small pool of public IP addresses in the area border gateway.

Transit networks

Transit VLAN IDs range from 2 to 254 and from 1000 to 1254 A /30 or /29 network is assigned to each transit VLAN, depending on the number of routers in each access node. The network prefix is identified as follows: Let v be the transit VLAN ID:

- If $v < 255$ the network is 10.v.0.0/30 or 10.v.0.0/29
- If $v > 1000$ the network is $10.(v - 1000).64.0/30$ or $10.(v - 1000).64.0/29$

Area border gateways is configured with the lowest address of the network while access point routers with the remaining ones in sequential order.

Access networks

User CPEs addresses are assigned from a /24 network that depends on the access radio to which it is connected. The network is identified as follows Let v be the transit VLAN ID:

- If $v < 255$ the network $10.(v).0.0/18$ is allocated to the access point
- If $v > 1000$ the network $10.(v - 1000).64.0/18$ is allocated to the access point.
- Access radios of the access point are enumerated starting from 1 to n, where n is the number of access radios installed. Being r the id of the radio, the IP addresses assigned to users connected to it falls in the $10.v.r.0/24$ or $10.(v - 1000).(64 + r).0/24$ network, depending on the conditions specified before.

In order to access the internet, private IP addresses assigned to users are mapped to public ones through CGNAT (Carrier grade network address translation). Usually a single /18 network is mapped to a single public IP, with some exceptions in case of large access point, where more than one public IP address is used, or in case

of small ones where multiple access points are mapped to the same public IP. On average, 50 private IP addresses are mapped to a single public address.

3.4.6 Traffic pattern

The network is used to provide fixed wireless access to residential customers, each one with an available bandwidth of 50 Mbit/s in downlink and 8 Mbit/s in uplink. The traffic pattern is the one typical of residential internet access, with prevalent downlink and a peak traffic in the evening. At peak hours, downlink traffic is about 10 times higher than uplink

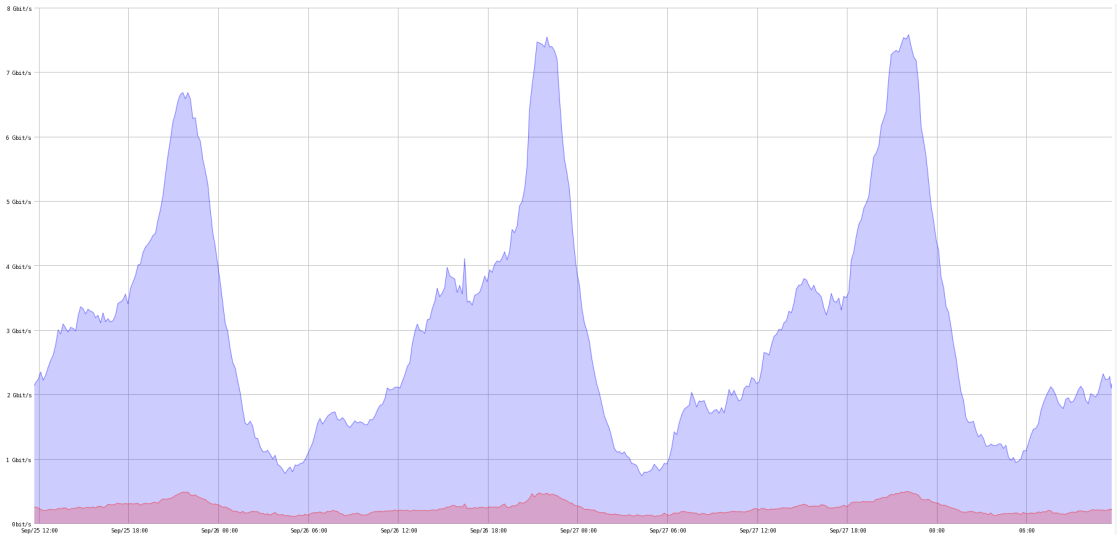


Figure 3.10: Network traffic in the Saluggia border gateway

Chapter 4

Layer 1 network improvements

Users demand for bandwidth has grown steadily over the years and the old transport network was no longer enough to satisfy it. Shortly before the beginning of this thesis project, a set of improvements were implemented in the network thanks to the liberalization of 60 GHz frequency bands and the release of new network devices

4.1 Implementation of new wireless bridges based on the IEEE 802.11ay/ax standards

The old transport network was mainly composed of these types of wireless bridges:

- **Ubiquiti AF24:** A bridge used to create high capacity links between large transit nodes.
- **Ubiquiti AF-5XHD:** A bridge used in medium capacity links to bridge medium-sized transit nodes. In a small number of nodes this device is coupled with an AF24
- **Ubiquiti PowerBeam 5AC 400:** A bridge used to connect access nodes and to create medium or low capacity transport links

The main problems of the old infrastructure were the unreliable capacity of PowerBeams due to the lack of an advanced RF filtering system and the insufficient capacity of AF-5XHD when used in transport links which caused congestion during peak hours in some parts of the network. To solve this issues, different solutions were implemented:

- Replacement of all the PowerBeams with new generation radios like AF-60, AF-5XHD, AF-24 and LTU LR to create links with a higher and more predictable capacity, especially when using devices operating in the 60 GHz and 24 GHz bands
- Replacement of AF-24 bridges with AF-60 and, when possible, creation of an additional link on the same path using devices operating in a different frequency band
- Replacement of AF-5XHD bridges with AF-24 or AF-60 if the available capacity is not enough
- When possible, shift towards a meshed topology with short distance, high capacity links

The implementation of new generation wireless bridges allowed us to increase bandwidth availability and stability across the entire network, especially when using AF-60 and AF-24 bridges which are almost immune to interference.

4.2 Implementation of new routers in large transport nodes

As the size of the network increased, the number of radios in large nodes increased as well and the Mikrotik CRS112-8P-4S switch seemed to be the best solution to connect all of them, but as the number of this kind of device in the network increased, some problems arose:

- It features a low-end CPU (single core, MIPS architecture, 450 MHz) which caused bottlenecks in access nodes where L3 routing is required between users PPPoE tunnels and the access node VLAN
- It stops forwarding traffic at random intervals requiring a manual reboot to restore its functionality

Since, on average, there was a failure every 24 hours, the user experience was heavily impacted and the Mikrotik CRS318-16P-2S+ was chosen as a replacement in all the locations because of its characteristics and low cost:

- A more powerful ARM CPU running at 750 MHZ capable of routing about 700 Mbit/s of IP traffic
- 16 Ethernet ports with PoE out support
- IP54 certification that allows to install the device in outdoor environments

- A simpler VLAN filtering configuration with respect to the CRS112
- Better value for money than the CRS112

About 80 CRS318 was installed which solved all the problems caused by the old switches: some units have been running for over an year without needing to be restarted and the bottlenecks in large access nodes were solved.

4.3 Migration to a new upstream provider

A few months before the beginning of this thesis project, the association decided to change the upstream provider in all the fiber optic switching nodes. The new one proposed two different ways of connecting our network to the internet:

1. Similarly to the current configuration, the provider establishes four different BGP sessions, one for each fiber optic switching node, leaving the network areas completely independent and isolated.
2. The new provider hosts a router owned and managed by the association in its datacenter and establishes with it a single BGP session. The provider also provisions four fiber optic layer 2 links between our datacenter router and the existing fiber opting switching points. A bandwidth of 20 Gbit/s is available in the datacenter while each fiber optic link has a maximum capacity of 10 Gbit/s

The second solution was preferred over the first one because it gave us greater flexibility: being the datacenter router the only point of interconnection with the internet in the entire network, the four network areas are no more isolated and new network configurations are possible:

- The NAT function could be moved in the datacenter router where the entire /22 IP range owned by the association could be announced, leading to a more efficient allocation of public IP addresses.
- Being the four network areas part of a single, large WAN, links between nodes of different areas are now possible.
- The available bandwidth in the datacenter can be dynamically allocated to each network area depending on the number of users
- Creation of a core network composed by the datacenter router and all those directly connected via fiber optics which can be expanded by provisioning new fiber optic links if needed

The main downside of this solution is that the colocated router is a single point of failure. In case of failures in the datacenter equipment, the entire network of the association would go offline so a backup router must be present and ready to be replaced to the main one in case of need.

The migration process is complex and it requires a large amount of time because new optical fiber links must be created. Currently, only the Sorin and Casale areas have been migrated to the new upstream provider so the advantages of the new architecture can only be exploited in this two areas

Chapter 5

Problems of the current network configuration and desired functionalities

In this chapter the problems and limitations of the current network design will be analyzed and a set of desired functionalities will be presented.

5.1 Loops handling

The main problem of the current network configuration in the lack of an efficient handling of network loops

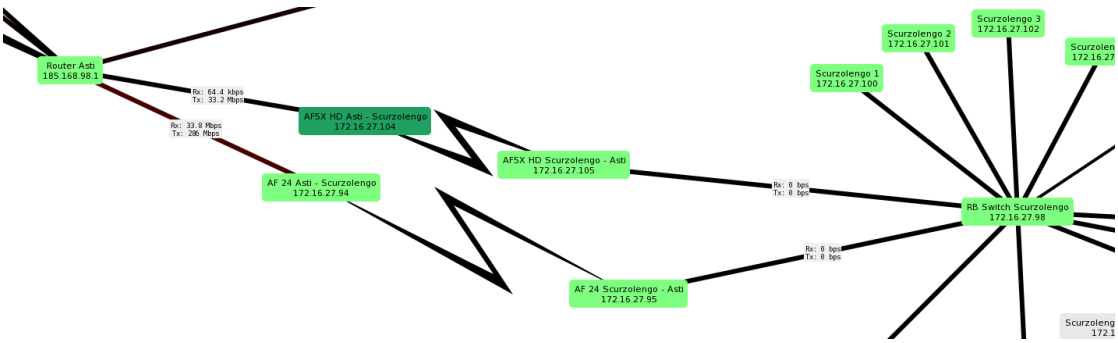


Figure 5.1: A network loop where a 24 GHz bridge is coupled with a 5 GHz one

With an increased number of wireless bridges operating in the 24 GHz and 60 GHz bands and higher bandwidth requirement the number of loops in the network increased. Currently network traffic is split across different branches of the loop

by filtering VLAN frames on different ports, which is a very simple and effective approach, but it present some issues:

- VLAN routes are fixed: in case of faults, network traffic is not automatically rerouted
- All the network hops are transparent from the IP point of view: it is impossible to use the traceroute utility for troubleshooting

5.2 Lack of stable IPv6 support

IPv6 support is experimental and must me manually configured for each user upon request. Additionally, there is no widespread IPv6 transport network

5.3 Lack of dynamic routing protocols

By using VLANs in the transport network to isolate traffic flows, VLAN filtering must be manually configured in each switch along the path. When a new access point is activated or the topology changes, all the involved switches must be manually reconfigured leading to a work overhead for network operators and possible errors in the switches configurations

5.4 Desired functionalities

The following is a list of functionalities that are needed to fully exploit all the network improvements described in chapter 4:

- **Automatic routes configuration in the transport network:** In order to automate the configuration process of the network routers and to avoid outdated and inconsistent configurations, dynamic protocols must be implemented which allow to automatically define routes in transit devices
- **Fault tolerance:** The new design must be fault tolerant. In case of link failures, traffic must be re-routed on the first available link with the highest priority
- **Full control over routing decisions:** Multiple network destinations may associate different priorities to the same set of links in order to fully exploit all the available bandwidth and to avoid link congestion in case of failures
- **Dual stack IPv4/IPv6 network:** An IPv6 transport and access network must be implemented and a suitable IPv6 addressing plan must be identified.

- **Support of links with different capacity:** Radio links capacity depends on the used technology and environmental conditions: load balancing techniques which equally share the available traffic on the available paths can not be used
- **Compatibility with existing network infrastructure:** Different kind of devices exist in the network with different characteristics and performances, and they can't be replaced due to the high cost involved. Most of the devices in the transport network are switches with limited L3 capabilities and CPU performance: the new design must adapt to this situation.
- **Low management overhead:** The new network configuration must be easy to modify in case of changes to the network

Chapter 6

Design of a new network configuration

This chapter describes all the network designs and protocols that have been analyzed and tested with the goal of finding one that is able to satisfy all the requirements.

6.1 Migration of the transport network from L2 to L3

In the old network design, from the IP point of view, the access point VLAN acted like a tunnel hiding the hops between the area border gateway and the access point router. Moreover, since the STP was disabled, re-routing was impossible. In order to re-route traffic in case of need, the IP layer must be aware of the hops between the source and destination: the transport network must be converted from a L2 network to an IP network.

6.2 Test environment

In the early testing phase, the GNS3 software has been used with RouterOS instances virtualized using the QEMU emulator. Only IPv4 connectivity is tested since the IPv6 network design is very similar. A simple network topology is used: the goal of this tests is solely to identify protocols that are able to satisfy design requirements. Once the best design is identified, more in-depth testing will be carried out in the real network

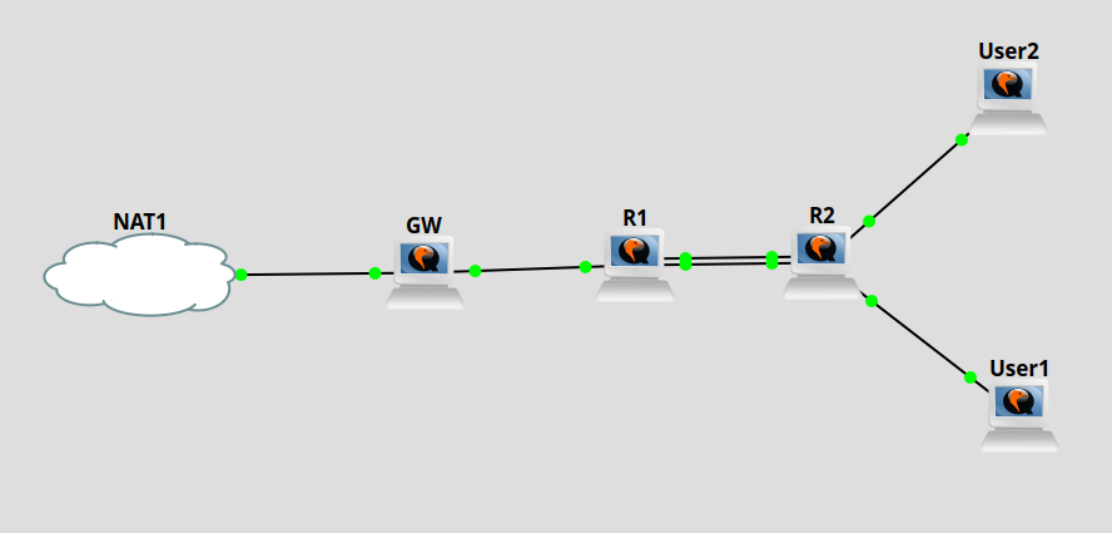


Figure 6.1: The GNS3 test topology

- **NAT1:** All packets routed to this cloud are forwarded to the public internet. Source NAT is performed and the new source address is the one of the machine which hosts GNS3. 192.168.122.1 is the next hop towards the internet

- **GW:** It represents the area border gateway. Internet access in provided via a default route with the NAT cloud as next hop. It also performs NAT by replacing the source IP address of the users with the one configured on the interface towards the Internet cloud.

- **R1:** A transit router.

- **R2:** An access point router. 10.0.0.0/24 and 10.0.1.0/24 are the access IP networks. It is the default gateway of the network of the users

- **User1 and user2:** Routers which simulate a user CPE. They are configured with 10.0.0.2 and 10.0.1.2 IP addresses respectively

A redundant link is present between R1 and R2. On each link, a /30 IP network in the 172.23.0.0/16 range is configured to provide point to point IP connectivity. The goal of the tests is to provide internet access to the devices in the 10.0.0.0/8 network satisfying all the requirements specified in section 4.2

6.3 Tested designs

6.3.1 Static routing with ECMP

This network design uses static routes coupled with ECMP (Equal cost multipath routing) to provide fault tolerance and load balancing between redundant links. ECMP implementation in RouterOS also supports links with different capacities but in this test an equal capacity on all the links is assumed

Devices configuration

This design is really straightforward: device configurations are self explanatory

```
#GW configuration:

/ip address
add address=192.168.122.2/24 interface=ether1 network=192.168.122.0
add address=172.23.0.1/30 interface=ether8 network=172.23.0.0
/ip firewall nat
add action=masquerade chain=srcnat src-address=10.0.0.0/8
/ip route
add dst-address=0.0.0.0/0 gateway=192.168.122.1
add dst-address=10.0.0.0/8 gateway=172.23.0.2
```

```
#R1 configuration:

/ip address
add address=172.23.0.2/30 interface=ether1 network=172.23.0.0
add address=172.23.0.5/30 interface=ether7 network=172.23.0.4
add address=172.23.0.9/30 interface=ether8 network=172.23.0.8
/ip route
add dst-address=0.0.0.0/0 gateway=172.23.0.1
add check-gateway=ping distance=1 dst-address=10.0.0.0/24 gateway
    =172.23.0.6
add check-gateway=ping distance=1 dst-address=10.0.0.0/24 gateway
    =172.23.0.10
add check-gateway=ping distance=1 dst-address=10.0.1.0/24 gateway
    =172.23.0.6
add check-gateway=ping distance=1 dst-address=10.0.1.0/24 gateway
    =172.23.0.10
```

```
#R2 configuration:
```

```
/ip address
add address=172.23.0.6/30 interface=ether2 network=172.23.0.4
add address=172.23.0.10/30 interface=ether1 network=172.23.0.8
add address=10.0.0.1/24 interface=ether8 network=10.0.0.0
add address=10.0.1.1/24 interface=ether7 network=10.0.1.0
/ip route
add check-gateway=ping distance=1 dst-address=0.0.0.0/0 gateway
    =172.23.0.5
add check-gateway=ping distance=1 dst-address=0.0.0.0/0 gateway
    =172.23.0.9
```

```
#Users configuration:

#User1:
/ip address add address=10.0.0.2/24 interface=ether1
/ip route add dst-address=0.0.0.0/0 gateway=10.0.0.1

#User2:
/ip address add address=10.0.0.2/24 interface=ether1
/ip route add dst-address=0.0.0.0/0 gateway=10.0.0.1
```

Verification and testing

A traceroute started in User1 shows that end users can reach the internet. Packets travel on different paths based on the destination addresses, showing that ECMP is also working. This design also allows to choose different paths for different IP destinations since the routing tables are under the full control of network operators

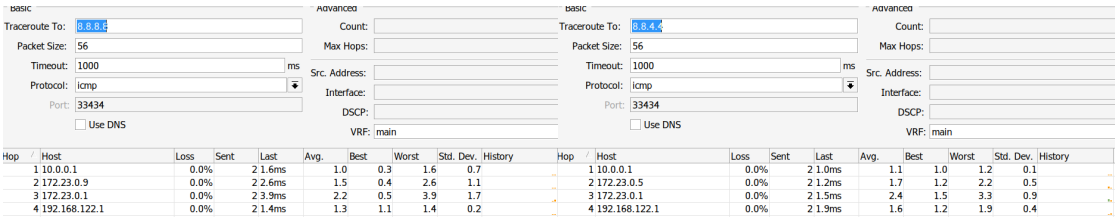


Figure 6.2: A traceroute started in User1. All the links are working

By disabling one of the redundant links, users are still able to reach the internet without any intervention from the network operator because the *check-gateway=ping* option detects a failure in the link. Packets always travel on the same path since there is only one left

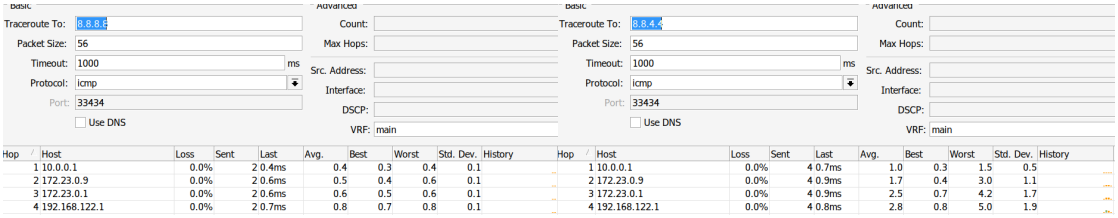


Figure 6.3: A traceroute started in User1. One of the links between R1 and R2 have been disabled

Conclusions

Among the strengths of this design, we primarily find robustness and simplicity, but the main drawback is that network operators must manually configure routes in each router along the path: in simple topologies like the testing one this is not a problem, but as the number of hops increases the possibility of errors and inconsistencies in routing tables increases as well, especially in case of topology changes where all the routing tables of the involved routers must be reconfigured. Another problem is caused by the variable capacity of wireless links: since the split ratio is static and must be manually configured, traffic may be unbalanced with respect to actual link capacities

6.3.2 OSPF

This design uses the OSPF protocol to populate routing tables. For testing purposes, the same topology is used and the network is configured with a backbone and a single not-so-stubby-area (NSSA). The backbone area contains the border gateway and the area border routers (ABR) while all the routers in the same branch or loop which originates from the backbone area belong to the same NSSA (R1 and R2 in the testing environment). ABRs (R1 in the testing environment) inject the default route in the NSSA and exports the NSSA routes (user networks) in the backbone area. NSSA was chosen because routers in a branch don't need to know detailed information about routes in another one: only a default route is needed and intra-area routing, if necessary, is performed by the ASBR. On the other hand, NSSA allows to export routes to the backbone area, which are needed by the GW to reach user networks. ECMP is also available in OSPF but the split ratio can't be modified.

Devices configuration

The IP address configuration is the same used in the previous design. All the static routes have been removed.

```
GW configuration
Roles: Autonomous system boundary router, backbone router

/routing id add id=10.254.0.1 name=id-1
/routing ospf instance add name=instance originate-default=if-installed router-id=id-1
/routing ospf area add instance=instance name=backbone
/routing ospf interface-template add area=backbone interfaces=ether8 type=ptp
```

```
R1 routing protocols configuration
Roles: Area border router, backbone router

/routing id add id=10.254.0.2 name=id-1
/routing ospf instance add name=instance originate-default=if-installed router-id=id-1
/routing ospf area
add instance=instance name=backbone
add area-id=0.0.0.1 instance=instance name=area1 type=nssa
/routing ospf interface-template
add area=backbone disabled=no interfaces=ether1 type=ptp
add area=area1 disabled=no interfaces=ether7,ether8 type=ptp
```

```
R2 routing protocols configuration
Roles: Internal router

/routing id add id=10.254.0.2 name=id-1
/routing ospf instance add name=instance originate-default=if-installed router-id=id-1
/routing ospf area
add instance=instance name=backbone
add area-id=0.0.0.1 instance=instance name=area1 type=nssa
/routing ospf interface-template
add area=area1 disabled=no interfaces=ether1,ether2 type=ptp
```

Verification and testing

	Dst. Address	Gateway	Distance
AS	0.0.0.0/0	192.168.122.1	1
DAo	10.0.0.0/24	172.23.0.2%ether8	110
DAo	10.0.1.0/24	172.23.0.2%ether8	110
DAC	172.23.0.0/30	ether8	0
DAo	172.23.0.4/30	172.23.0.2%ether8	110
DAo	172.23.0.8/30	172.23.0.2%ether8	110
DAC	192.168.122.0/..	ether1	0

Figure 6.4: GW routing table

	Dst. Address	Gateway	Distance
DAo	0.0.0.0/0	172.23.0.1%ether1	110
DAo+	10.0.0.0/24	172.23.0.10%ether8	110
DAo+	10.0.0.0/24	172.23.0.6%ether7	110
DAo+	10.0.1.0/24	172.23.0.10%ether8	110
DAo+	10.0.1.0/24	172.23.0.6%ether7	110
DAC	172.23.0.0/30	ether1	0
DAC	172.23.0.4/30	ether7	0
DAC	172.23.0.8/30	ether8	0

Figure 6.5: R1 routing table

	Dst. Address	Gateway	Distance
DAo+	0.0.0.0/0	172.23.0.5%ether2	110
DAo+	0.0.0.0/0	172.23.0.9%ether1	110
DAC	10.0.0.0/24	ether8	0
DAC	10.0.1.0/24	ether7	0
DAo+	172.23.0.0/30	172.23.0.5%ether2	110
DAo+	172.23.0.0/30	172.23.0.9%ether1	110
DAC	172.23.0.4/30	ether2	0
DAC	172.23.0.8/30	ether1	0

Figure 6.6: R2 routing table

All the routing tables are correctly populated: user routes are exported in the backbone area and the default route is injected in the NSSA. Users are able to reach the internet and traffic takes different paths depending on the destination. In case of failures on one link, traffic is automatically re-routed on the backup one.

Hop	Host	Loss	Sent	Last	Hop	Host	Loss	Sent	Last
1	10.0.1.1	0.0%	2	0.3ms	1	10.0.1.1	0.0%	2	1.6ms
2	172.23.0.5	0.0%	2	0.5ms	2	172.23.0.9	0.0%	2	2.9ms
3	172.23.0.1	0.0%	2	0.6ms	3	172.23.0.1	0.0%	2	4.2ms
4	192.168.122.1	0.0%	2	0.6ms	4	192.168.122.1	0.0%	2	5.2ms
5	192.168.100.1	0.0%	2	1.9ms	5	192.168.100.1	0.0%	2	6.6ms
6	130.192.164.254	0.0%	2	2.4ms	6	130.192.164.254	0.0%	2	3.2ms
7	130.192.232.60	0.0%	2	1.9ms	7	130.192.232.60	0.0%	2	2.2ms
8	130.192.232.254	0.0%	2	4.6ms	8	130.192.232.254	0.0%	2	2.8ms
9	193.206.132.34	0.0%	2	2.1ms	9	193.206.132.34	0.0%	2	18.0ms
10	185.191.181.129	0.0%	2	6.3ms	10	185.191.181.127	0.0%	2	3.5ms
11	185.191.181.35	0.0%	2	11.2ms	11	185.191.181.119	0.0%	2	4.7ms
12	185.191.181.69	0.0%	2	18.9ms	12	185.191.180.157	0.0%	2	11.6ms
13	142.250.164.230	0.0%	2	19.0ms	13	142.250.174.46	0.0%	2	5.9ms
14	108.170.245.65	0.0%	2	20.1ms	14	72.14.239.144	0.0%	2	5.3ms
15	142.251.235.173	0.0%	2	12.7ms	15	142.250.211.31	0.0%	2	4.2ms
16	8.8.8.8	0.0%	1	12.8ms	16	8.8.4.4	0.0%	2	11.4ms

Figure 6.7: A traceroute started in User1 shows that ECMP routing is correctly working by balancing network traffic among different links

	Dst. Address	Gateway	Distance
DAo	0.0.0.0/0	172.23.0.1%ether1	110
DAo	10.0.0.0/24	172.23.0.6%ether7	110
DAo	10.0.1.0/24	172.23.0.6%ether7	110
DAC	172.23.0.0/30	ether1	0
DAC	172.23.0.4/30	ether7	0
DAC	172.23.0.8/30	ether8	0

Figure 6.8: R1 routing table when the ether1 interface on R2 is disabled, simulating a failure on one of the two links between R1 and R2

Conclusions

The dynamic neighbor discovery function provided by OSPF and the dynamic population of routing table make this design more scalable and less prone to errors than the previous one. IP routing hardware offloading is also supported but not all the requirements are satisfied. The OSPF protocol behaves similarly to the spanning tree: in presence of loops, 2 behaviours are possible:

1. If all the links have the same cost, traffic is equally split among them, leading to unbalanced traffic flows in case of different capacity links
2. If link costs are different, traffic is routed only through the best one while all the other links are kept disabled, wasting all the available bandwidth in other links

This design, although better than the previous one from the scalability point of view, is not applicable in the network due to the problems listed above

6.3.3 MPLS

MPLS[10], especially in its traffic engineering variant, presents a series of useful features:

- Possibility to specify the path towards a specific destination hop-by-hop
- Possibility to define backup links in case of failure of the main one. The *Fast-Reroute* path, installed together with the main one, may be used within 50 milliseconds after the failure of the main one, guaranteeing a seamless migration on the backup
- Possibility to specify link capacities and to install LSPs accordingly

Unfortunately, in Mikrotik devices MPLS hardware forwarding is still not supported [11], requiring the CPU to perform all the MPLS operations. In the CRS318-16P-2S+ switch, which is the one used in main transport nodes with an aggregated throughput up to 3 Gbit/s, the channel between the switching chip and the CPU has a maximum half duplex capacity of 1,3 Gbit/s, making this switch unsuitable in many transport nodes

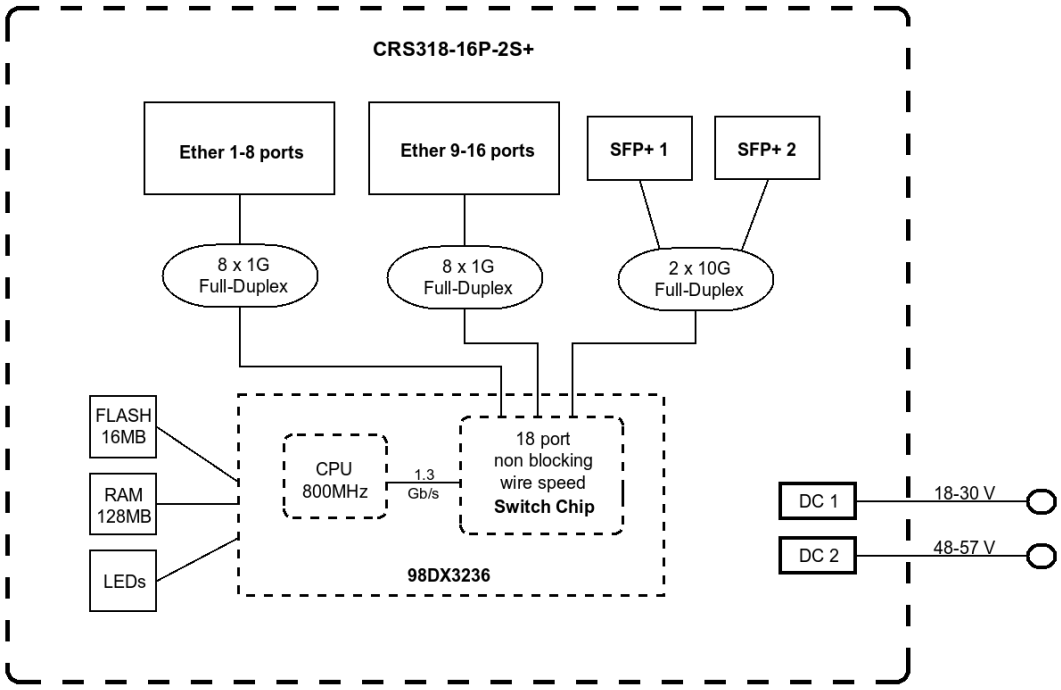


Figure 6.9: CRS318-16P-2S+ block diagram [12]

6.3.4 BGP and OSPF

BGP is an exterior gateway protocol usually used to exchange routing information between different autonomous systems. However, thanks to its large set of settings which allows a fine tuning of the routes announcements and selection procedure, it can also be used as an IGP to perform some basic traffic engineering operations. In this test, each node is treated as an autonomous system announcing its access network and serving as a transit for other nodes: a private ASN equal to the VLAN ID is assigned to each access/transit node while an incremental number starting from 2000 is assigned to pure transit nodes which doesn't have a VLAN ID assigned. The BGP session between peers is established over a /30 point to point IP network extracted from the 172.23.0.0/16 range, configured on each link. BGP is used by access routers to announce their user networks and by area border gateways to announce the default route, while OSPF is used to announce P2P networks.

Selection of the preferred link

BGP sessions are established between neighbor devices using P2P addresses and multiple BGP sessions between the same peers are possible in case multiple links are available. Networks are announced on each BGP session, thus on each link: in order to specify priorities when multiple paths are available, routing filters are used coupled with address lists. The following is a description of the adopted solution, which is applied in each router that has the possibility to reach an access point network through more than one link:

1. If an access point network can be reached from n different links, then each link has n different levels of priority.

2. Within the scope of a specific link, each level of priority is associated with an address list, whose name is *n-name*: n is the priority and *name* is the name of the link

3. Networks are inserted in the address lists according to the desired link priority. For example, if the network 10.0.0.0/24 is reachable from *link1* and *link2* with link1 being preferred over link2, it will be inserted in the lists *1-link1* and *2-link2*

4. For each link, and therefore BGP session, the following chain of routing filters is matched against incoming announcements:

```
1    # Routing filters chain for link1
2
3    # Rule 1
```

```
4     if(destination in 1-link1) then {set bgp-local-pref 200-1;
      accept;}
5
6     # Rule 2
7     if(destination in 2-link1) then {set bgp-local-pref 200-2;
      accept;}
8
9     # Rule n
10    if(destination in n-link1) then {set bgp-local-pref 200-n;
      accept;}
11
12    # Rule n+1 (fallback rule in case the announced network is
      not found in any address list)
13    set bgp-local-pref 100-1; accept;
14
```

```
1     # Routing filters chain for link2
2
3     # Rule 1
4     if(destination in 1-link2) then {set bgp-local-pref 200-1;
      accept;}
5
6     # Rule 2
7     if(destination in 2-link2) then {set bgp-local-pref 200-2;
      accept;}
8
9     # Rule n
10    if(destination in n-link2) then {set bgp-local-pref 200-n;
      accept;}
11
12    # Rule n+1 (fallback rule in case the announced network is
      not found in any address list)
13    set bgp-local-pref 100-2; accept;
14
```

5. Routing filters, upon the reception of a route announcement, are evaluated from the first to the last one of the chain: when a match is found because the destination network is found in the address list, the corresponding BGP local preference is applied to the route announcement which is later used by the BGP selection algorithm: routes with a higher local preference are preferred over lower ones without taking into consideration the AS path length and other conditions.

6. In case of failure of the active link, the corresponding BGP session drops and the second route with the highest local preference is selected and installed in

the routing table

If a router receives a default route announcement from n different upstreams, then n priorities are defined. Each priority is identified by a routing filters chain:

```
# Routing filters chain for priority 1
#Rule 1
if(dst == 0.0.0.0/0) then {set bgp-local-pref 200-1; accept;} else {
    accept;}

# Routing filters chain for priority 2
#Rule 1
if(dst == 0.0.0.0/0) then {set bgp-local-pref 200-2; accept;} else {
    accept;}

# Routing filters chain for priority n
#Rule 1
if(dst == 0.0.0.0/0) then {set bgp-local-pref 200-n; accept;} else {
    accept;}
```

This rules accept all the incoming announcements with the local preference being modified according to the priority if the destination is the default route. Routing filters chains are associated to BGP sessions depending on the desired priority. In case of link failures, another default route is selected with the same mechanism used for access point networks. A downside of this solution is that all the uplink traffic of a loop is forced over the same link while the downlink may take different paths: this is not a problem because, as explained in subsection 3.4.6, the uplink traffic is about 10 times lower than the downlink one and forcing it on high capacity links (Ubiquiti Airfiber 60 LR or Ubiquiti Airfiber 24) is enough to provide sufficient uplink capacity.

Devices configuration

Link 1 (network 172.23.0.4/30) carries the traffic directed towards the default route and 10.0.0.0/24 while link 2 (network 172.23.0.8/30) towards 10.0.1.0/24. Hold time and keep-alive time are 10 seconds and 2 seconds respectively: this ensures that a link failure is detected within a maximum of 10 seconds. This values can be lowered if faster reaction times are needed. OSPF configuration is the same as before with the only two differences being the type of the area which is now stub because there is no need to export in the backbone area the access routes and the default route injection in the area, which is now disabled: these tasks are performed by BGP. For the sake of brevity, only BGP and related configurations are reported

```
1  # GW configuration
2  /routing bgp template set default address-families=ip as=1 hold-
   time=10s keepalive-time=2s output.default-originate=if-installed
   router-id=10.254.0.1 routing-table=main
3  /routing bgp connection add local.role=ebgp name=to-r1 remote.
   address=172.23.0.2/32 templates=default
```

```
1  # R1 configuration
2
3  /ip firewall address-list
4  add address=10.0.0.0/24 list=1-link1
5  add address=10.0.0.0/24 list=2-link2
6  add address=10.0.1.0/24 list=1-link2
7  add address=10.0.1.0/24 list=2-link1
8
9  /routing filter rule
10 add chain=bgp-in-link1 rule="if(dst in 1-link1) {set bgp-local-
   pref 199; accept;}"
11 add chain=bgp-in-link1 rule="if(dst in 2-link1) {set bgp-local-
   pref 198; accept;}"
12 add chain=bgp-in-link1 rule="set bgp-local-pref 99; accept;"
13 add chain=bgp-in-link2 rule="if(dst in 1-link2) {set bgp-local-
   pref 199; accept;}"
14 add chain=bgp-in-link2 rule="if(dst in 2-link2) {set bgp-local-
   pref 198; accept;}"
15 add chain=bgp-in-link2 rule="set bgp-local-pref 98; accept;"
16
17 /routing bgp template
18 set default address-families=ip as=2 disabled=no hold-time=10s
   keepalive-time=2s output.default-originate=if-installed router-id
   =10.254.0.2 routing-table=main
19
20 /routing bgp connection
21 add disabled=no local.role=ebgp name=to-gw remote.address
   =172.23.0.1/32 templates=default
22 add address-families=ip as=2 disabled=no hold-time=10s input.
   filter=bgp-in-link2 keepalive-time=2s local.role=ebgp name=to-r2-
   link2 output.default-originate=if-installed remote.address
   =172.23.0.10/32 router-id=10.254.0.2 routing-table=main templates=
   default
23 add address-families=ip as=2 disabled=no hold-time=10s input.
   filter=bgp-in-link1 keepalive-time=2s local.role=ebgp name=to-r2-
   link1 output.default-originate=if-installed remote.address
   =172.23.0.6/32 router-id=10.254.0.2 routing-table=main templates=
   default
```

```
# R2 configuration

/ip firewall address-list
add address=10.0.0.0/24 list=bgp-networks
add address=10.0.1.0/24 list=bgp-networks

/routing filter rule
add chain=bgp-in-default1 disabled=no rule="if(dst == 0.0.0.0/0)
{set bgp-local-pref 199; accept;} else {accept}"
add chain=bgp-in-default2 disabled=no rule="if(dst == 0.0.0.0/0)
{set bgp-local-pref 198; accept;} else {accept}"

/routing bgp template
set default address-families=ip as=3 disabled=no hold-time=10s
keepalive-time=2s router-id=10.254.0.3 routing-table=main

/routing bgp connection
add address-families=ip as=3 disabled=no hold-time=10s input.
filter=bgp-in-default2 keepalive-time=2s local.role=ebgp name=to-
r1-link2 output.network=bgp-networks remote.address=172.23.0.9/32
router-id=10.254.0.3 routing-table=main templates=default
add address-families=ip as=3 disabled=no hold-time=10s input.
filter=bgp-in-default1 keepalive-time=2s local.role=ebgp name=to-
r1-link1 output.network=bgp-networks remote.address=172.23.0.5/32
router-id=10.254.0.3 routing-table=main templates=default
```

Verification and testing

Routing tables of R1 and R2 shows that the local preference is correctly applied to announces and the highest one is selected. A traceroute from user1 and user2 shows that network traffic is transmitted on the desired links

	Dst. Address	Gateway	Distance
AS	0.0.0.0/0	192.168.122.1	1
DAb	10.0.0.0/24	172.23.0.2	20
DAb	10.0.1.0/24	172.23.0.2	20
DAC	172.23.0.0/30	ether8	0
DAo	172.23.0.4/30	172.23.0.2%ether8	110
DAo	172.23.0.8/30	172.23.0.2%ether8	110
DAC	192.168.122.0/..	ether1	0

Figure 6.10: GW routing table

	Dst. Address	Immediate Gateway	Local Pref.
DAb	0.0.0.0/0	172.23.0.1%ether1	
Db	10.0.0.0/24	172.23.0.10%ether8	198
DAb	10.0.0.0/24	172.23.0.6%ether7	199
DAb	10.0.1.0/24	172.23.0.10%ether8	199
Db	10.0.1.0/24	172.23.0.6%ether7	198
DAC	172.23.0.0/30	ether1	
DAC	172.23.0.4/30	ether7	
DAC	172.23.0.8/30	ether8	

	Dst. Address	Immediate Gateway	Local Pref.
Db	0.0.0.0/0	172.23.0.9%ether1	198
DAb	0.0.0.0/0	172.23.0.5%ether2	199
DAC	10.0.0.0/24	ether8	
DAC	10.0.1.0/24	ether7	
DAo+	172.23.0.0/30	172.23.0.5%ether2	
DAo+	172.23.0.0/30	172.23.0.9%ether1	
DAC	172.23.0.4/30	ether2	
DAC	172.23.0.8/30	ether1	

Figure 6.11: a) Routing table of R1 and b) Routing table of R2

Hop	Host	Loss	Sent	Last
1	10.0.0.1	0.0%	2	0.4ms
2	172.23.0.5	0.0%	2	0.8ms
3	172.23.0.1	0.0%	2	0.7ms
4	192.168.122.1	0.0%	2	0.6ms

Hop	Host	Loss	Sent	Last
1	10.0.1.1	0.0%	2	1.7ms
2	172.23.0.9	0.0%	2	2.6ms
3	172.23.0.1	0.0%	2	4.2ms
4	192.168.122.1	0.0%	2	3.2ms

Figure 6.12: a) Traceroute from User1 (10.0.0.2) and b) Traceroute from User2 (10.0.1.2)

By disabling link1, all the traffic is automatically rerouted to link2 within 10 seconds

	Dst. Address	Immediate Gateway	Local Pref.
DAb	0.0.0.0/0	172.23.0.1%ether1	
DAb	10.0.0.0/24	172.23.0.10%ether8	198
DAb	10.0.1.0/24	172.23.0.10%ether8	199
DAC	172.23.0.0/30	ether1	
DAC	172.23.0.4/30	ether7	
DAC	172.23.0.8/30	ether8	

	Dst. Address	Immediate Gateway	Local Pref.
DAb	0.0.0.0/0	172.23.0.9%ether1	198
DAC	10.0.0.0/24	ether8	
DAC	10.0.1.0/24	ether7	
DAo	172.23.0.0/30	172.23.0.9%ether1	
DAo	172.23.0.4/30	172.23.0.9%ether1	
DAC	172.23.0.8/30	ether1	

Figure 6.13: a) Routing table of R1 and b) Routing table of R2 when link1 is disabled

Conclusions

This design satisfies all the requirements. The only downsides are that each BGP session must be configured manually and the configuration is slightly more complex with respect to the previous ones. This design is also scalable: when an access node is added, removed or the topology changes, only the BGP sessions and, if needed, routing filters and address lists of the involved nodes must be updated. L3 hardware offloading is also possible, enabling to reach wire-speed when routing IP packets

6.4 The selected design

Given the test results and considerations, the candidate identified the *"BGP and OSPF"* design as the most suitable to be implemented in the existing network because of its scalability and flexibility. This design also allows to efficiently leverage existing hardware, particularly L3 hardware offloading.

6.5 New network configuration

In order to adapt the tested design to the physical network, which differs from the testing one in complexity and topology, some changes must be implemented and additional configuration parameters are needed

6.5.1 PTP VLANs

In older sections of the network, where the *switch + router* topology is typically used, switches without L3 capabilities are still present: this devices can't run the BGP protocol and route IP packet, so they must be transparent with respect to the IP transport network. A possible solution is to use VLANs to create virtual tunnels between RouterOS devices. L3 hardware offloading is also supported for inter-VLAN routing so this configuration does not decrease the performance of the network design. PTP VLAN IDs are defined in an incremental way starting from ID 3000. The name of the VLAN virtual interface in the router is *vlanN-upstreamDownstream*:

- N is the vlan ID
- upstream is the upstream node
- downstream is the downstream node

Note that BGP peers may be several hops distant in case they are connected by switches without L3 capabilities: the PTP VLAN acts as a tunnel hiding the underlying L2 network so the network devices that are part of a loop must be routers.

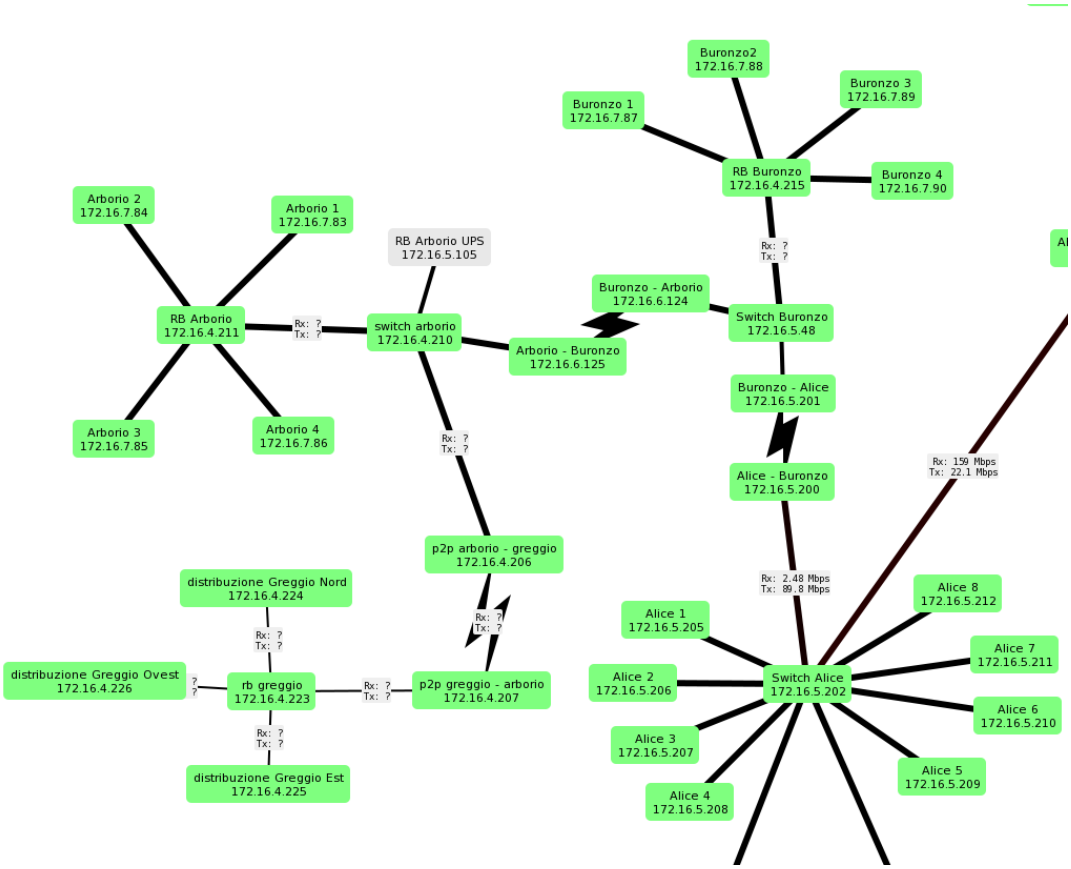


Figure 6.14: An old branch of the network were switches without L3 capabilities are still in use. PTP VLANs start in *Switch Alice* and are terminated in *RB Buronzo, RB Arborio, RB Greggio*

6.5.2 Transit IP networks

A /30 IP network is configured on each PTP VLAN to enable IP communication between routers. Each network is taken from a /23 or larger network which is the same for all the links in the same network branch with an area border gateway as root. These /23 or larger networks are extracted from the 172.23.0.0/16 address range.

6.5.3 IPv6 core network

Since the existing IPv6 transport network was experimental and used by a limited amount of users, the new one has been directly implemented in the physical network. The IPv6 network uses the same protocols and configuration strategies of the IPv4 one, with the main difference being the addressing scheme:

- The association owns the 2a0a:1300::/29 prefix. A /32 prefix extracted from that one is assegned to each network area

- A /64 network is configured in each PTP link, as suggested in [13]. The address of the upstream router is 2a0a:1301:f:vlanId::1 while the downstream one is 2a0a:1301:f:vlanId::2

  - *2a0a:1301* is the network area prefix
  - *vlanId* is the PTP VLAN id.

- Each access node announces a /44 network and each user receives a /56 prefix via prefix delegation. The format of the IPv6 prefix announced by the access node is 2a0a:1301:n0::/44 where:

  - *2a0a:1301* is the network area prefix
  - n=256+ASN if 0<ASN<=254
  - n=ASN-512 if 1000<ASN<=1254

The implementation details of the IPv6 access network is left for future work

6.5.4 Large access node with multiple routers

All the users connected to the same access node are assigned an IP address taken from the /18 network of the access node. In large access nodes, where more than one router is used to connect access radios, the large /18 network is split in smaller /24 networks, one for each access radio. Multiple solutions are possible:

- Create multiple BGP sessions between the routers and the upstream node

- Aggregate all the smaller networks in a single /18 network announced by a single router. PPPoE sessions are established between users and the router that is in charge of announcing the AP network. L2 domains between the router and access radios are isolated by VLANs.

The last solution has been identified as the most appropriate one: a single, larger network is announced in BGP sessions leading to a reduction in the size of routing tables, and it is also simpler to configure address lists in routers which are upstreams of network loops.

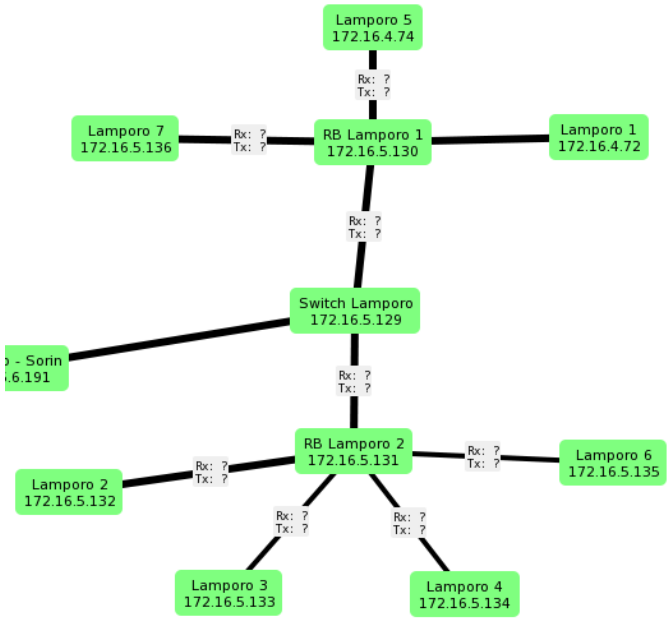


Figure 6.15: A large access node with multiple routers (*RB Lamporo 1* and *RB Lamporo 2*

6.5.5 Loopback IP

A loopback IP is an IP address configured on a virtual bridge interface with no ports attached. Since the virtual bridge never goes down, the loopback IP is always reachable from all the interfaces and can be considered as a reference for the router, uniquely identified by it in the network. In the new configuration, loopback IPs format is *10.ASN.0.2* and they are used for different purposes:

- IP endpoint for the access point DNS cache
- As a source IP for radius requests
- As a way to reach the router without using the management network.

6.5.6 L3 hardware offloading

CRS318-16P-2S+ switches are capable of offloading IP routing to the switching chip [11], allowing the device to reach near wire-speed when routing IP packets. In large transit nodes, where this switches are mainly deployed, L3 hardware offloading must be enabled because the channel between the switch chip and the CPU has a limited half-duplex capacity of 1,3 Gbit/s which is not enough in most situations.

L3 hardware offloading supports inter-VLAN routing, therefore PTP VLANs will not interfere with its behaviour

6.5.7 BGP and OSPF configuration in access/transit routers

The BGP and OSPF configuration is the same used in the tested design, with the main differences being the router ID, which is equal to the loopback IP, and the OSPF area ID which is the network ID of the /23 address range assigned to the OSPF area.

6.5.8 Area border gateways and datacenter router

Core routers configuration differs from the one of transit/access routers:

- **Datacenter router:** OSPF is not enabled. Two different independent BGP instances are configured on this router: one is for peering with the upstream provider and another is for peering with area border gateways. The first BGP instance receives a default route from the upstream provider and announces the public IP address range, while the latter redistributes the default route to area border gateways and receives the access network routes
- **Area border gateway:** A single BGP instance is running. Peering sessions are established with the datacenter router and each downstream router. Access network routes are received from downstream routers and a default route is announced to them. OSPF is running in this routers which, being the root of every branch or loop present in the network area, are members of all OSPF areas

Chapter 7

Deployment of the new network configuration

7.1 Deployment guidelines

The deployment of the new configuration in the physical network has been performed according to the following guidelines:

1. **Incremental rollout:** The reconfiguration process has been performed by following an incremental approach in order to allow network administrators to test the new design in small sections of the network: in case of problems, the number of impacted users is limited and it's easier to make corrections to the configuration because of the little number of devices involved.

2. **Coexistence of the new configuration with the old one:** Because of the incremental approach followed during the configuration, the whole process was expected to last a few months: during this time period, the new design had to coexist with the old one without conflicts

3. **Minimization of the downtime:** The reconfiguration should not impact the user experience: disconnections should possibly not occur but, if they do, their duration must be limited to a few minutes and the number of impacted users must be as small as possible

4. **Standardization of configurations:** The reconfiguration script applied in all the routers need to be generated starting from a common template in order to reduce the number of errors and inconsistencies while speeding up the process

7.2 Reconfiguration script

Mikrotik routers don't have the possibility to create an "offline" configuration which could be later installed: the only possibility to reconfigure routers is to edit the running configuration and each change is immediately applied. The reconfiguration script, which is shown in Appendix A, must take care of it. The script is made of different sections:

1. **Variables:** The first section contains the definition of the constants used by the configuration commands. The table shown in Appendix A contains the description of their values. Before executing the script in each router, these values must be manually edited with the correct ones

2. **Miscellaneous:** Loopback IP configuration and minor adjustments to the existing configuration

3. **PTP related configuration:** Configuration of the PTP VLANs and IPs, both IPv4 and IPv6. VLAN filtering is also configured

4. **Announced networks and routing filters:** Definition of the routing filter chains. Creation of the address lists containing the announced network and corresponding blackhole routes

5. **OSPF:** Configuration of the OSPF protocol to announce transit IP networks

6. **BGP:** Configuration of the BGP instance, upstream and, if any, downstream peers

7. **Firewall:** Firewall configuration of the router

7.3 First rollout

The first roll out has been performed in a small portion of the network with a similar topology to the tested one and involved two nodes: *Lauriano Scuola*, an hybrid node, and *Moriondo*, an access node

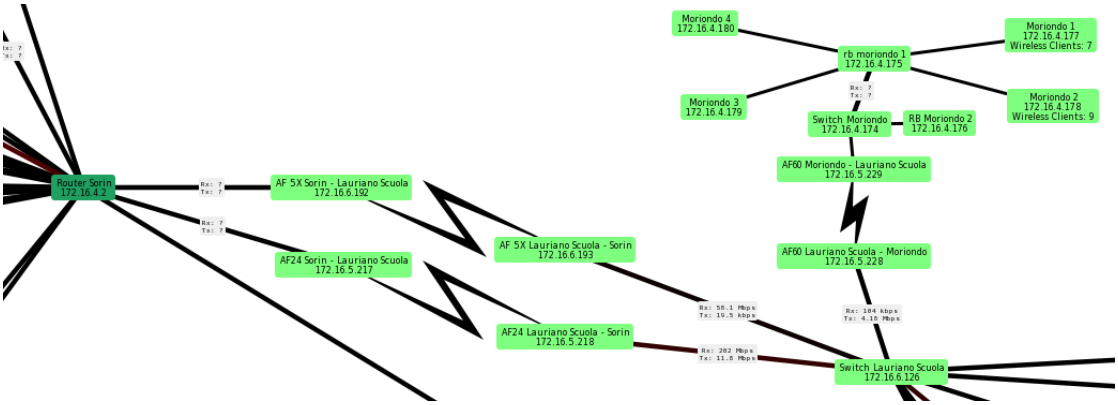


Figure 7.1: The first portion of the network that has been reconfigured. It contains a loop between the area border gateway (Router Sorin) and two routers connected in sequence (*Switch Lauriano Scuola* and *RB Moriondo 1*)

Network devices have been reconfigured starting from the datacenter router down to the last access router, *RB Moriondo 1*. The snippets showed in Appendix B contain the main sections of the new configuration. *Lauriano Scuola* is an hybrid node: it announces the network 10.180.0.0/18 and serves as a transit for *Moriondo*, which announces the network 10.35.0.0/18 Downlink traffic directed to *Lauriano scuola* must prefer the *Sorin - Lauriano 24* link while that directed to *Moriondo* must prefer *Lauriano - Sorin AFX*. Being *Sorin - Lauriano 24* a full-duplex, 700 Mbit/s link, it must be the preferred path for uplink traffic of both nodes

7.3.1 Verification and testing

By querying the routing tables of the datacenter and Sorin routers, we can observe that routes towards *Moriondo* and *Lauriano scuola* access networks are correctly installed. In the Sorin router, link priorities are correctly working and both links are being used. PTP network routes are also active, showing that the OSPF protocol is correctly working. The "+" flag assigned to the 172.23.0.8/30 route means that more than one path is available and that ECMP is equally splitting traffic across the available links: this behaviour may lead to unoptimized traffic paths in case of high amount of traffic towards PTP networks but the actual traffic is very low since these routes are only used for management purposes, RADIUS requests and DNS queries of the access node local cache

```
Flags: D - DYNAMIC; A - ACTIVE; b, y - BGP-MPLS-VPN
Columns: DST-ADDRESS, GATEWAY, DISTANCE
    DST-ADDRESS     GATEWAY     DISTANCE
DAb 10.35.0.0/18    172.16.0.2        20
DAb 10.180.0.0/18   172.16.0.2        20
```

```
Flags: D - DYNAMIC; A - ACTIVE; c, b, o, y - BGP-MPLS-VPN; + - ECMP
Columns: DST-ADDRESS, IMMEDIATE-GW
    DST-ADDRESS     IMMEDIATE-GW
DAb  0.0.0.0/0      172.16.0.1%vlan4000-UtentiSorin
DAb  10.35.0.0/18   172.23.0.6%vlan3002-SorinLaurianoAFX
D b  10.35.0.0/18   172.23.0.2%vlan3001-SorinLauriano24
D b  10.180.0.0/18  172.23.0.6%vlan3002-SorinLaurianoAFX
DAb  10.180.0.0/18  172.23.0.2%vlan3001-SorinLauriano24
DAc  172.23.0.0/30  vlan3001-SorinLauriano24
DAc  172.23.0.4/30  vlan3002-SorinLaurianoAFX
DAo+ 172.23.0.8/30  172.23.0.2%vlan3001-SorinLauriano24
DAo+ 172.23.0.8/30  172.23.0.6%vlan3002-SorinLaurianoAFX
```

Figure 7.2: Routes towards *Lauriano scuola* and *Moriondo* installed in the a) datacenter router and b) Sorin router

The backup mechanism is also working: by disabling the *Sorin - Lauriano AFX* link, all the traffic is routed through the other one

```
Flags: D - DYNAMIC; A - ACTIVE; c, b, o, y - BGP-MPLS-VPN
Columns: DST-ADDRESS, IMMEDIATE-GW
    DST-ADDRESS     IMMEDIATE-GW
DAb 0.0.0.0/0       172.16.0.1%vlan4000-UtentiSorin
DAb 10.35.0.0/18    172.23.0.2%vlan3001-SorinLauriano24
DAb 10.180.0.0/18   172.23.0.2%vlan3001-SorinLauriano24
DAc 172.23.0.0/30   vlan3001-SorinLauriano24
DAo 172.23.0.4/30   172.23.0.2%vlan3001-SorinLauriano24
DAo 172.23.0.8/30   172.23.0.2%vlan3001-SorinLauriano24
```

Figure 7.3: Routing table of the Sorin router when the *Sorin - Lauriano AFX* link is offline

In the router located in *Lauriano scuola* node, we can see that two default routes are available, but only the one received from the link *Sorin - Lauriano 24* is active, as expected. The route towards *Moriondo* access network is also installed and active. All the routes have the "H" flag, meaning that L3 hardware offloading is active and correctly working.

```
Flags: D - DYNAMIC; A - ACTIVE; c, s, b, y - BGP-MPLS-VPN; H - HW-OFFLOADED
Columns: DST-ADDRESS, IMMEDIATE-GW
#      DST-ADDRESS     IMMEDIATE-GW
  D bH 0.0.0.0/0       172.23.0.5%vlan3002-SorinLaurianoAFX
  DAbH 0.0.0.0/0       172.23.0.1%vlan3001-SorinLauriano24
  DAbH 10.35.0.0/18    172.23.0.10%vlan3008-LaurianoMoriondo
0  As  10.180.0.0/18
  DAcH 172.23.0.0/30   vlan3001-SorinLauriano24
  DAcH 172.23.0.4/30   vlan3002-SorinLaurianoAFX
  DAcH 172.23.0.8/30   vlan3008-LaurianoMoriondo
```

Figure 7.4: Routing table of the router located in *Lauriano scuola* node

Following the reconfiguration, the behaviour of this part of the network was carefully monitored for about 2 weeks, as well as user feedback, in order to detect any problems or erroneous behaviour and the only problem detected is relative to

the L3 hardware offloading: in case of frequent reconvergence of routing protocols caused, for example, by link drops due to rain, L3 hardware offloading stops working and a manual reboot of the device is needed. This is a known bug of the operating system which has been fixed in RouterOS 7.11.

During the testing period, some heavy thunderstorms occurred in the area and the *Sorin - Lauriano 24* link, which operates in the 24 GHz band, dropped for about 20 minutes: in less that 10 seconds all the traffic was diverted on the alternative link keeping downstream users online

7.4 Complete rollout

Given the positive outcome of the initial test, it was possible to reconfigure all the network area depending on the fiber optic switching point in Saluggia and then the entire network. To speedup the script generation process, a small Java program was written. It takes as input two CSV files and the script template:

- **AS CSV file:** It contains information about the reconfigured access nodes. Each row refers to a single access node with the following properties:
  - ASN
  - Router name
  - OSPF area ID
  - OSPF area ID prefix length
  - Access network of the node
  - IPv6 prefix of the network area
- **Links CSV file:** It contain information about the links between nodes and PTP connectivity. Each row refers to a single link with the following properties:
  - Link name
  - Upstream ASN
  - Downstream ASN
  - Upstream ethernet port
  - Downstream ethernet port
  - Upstream PTP address
  - Downstream PTP address
  - PTP Vlan ID

By combining data from CSV files, the program injects constant values in the template shown in Appendix B and produces an output file for each router contained in the *as.csv* file. In particular, by using the ASN value as a link between the two files, it is able to generate the list of upstream and downstream BGP sessions for each router, saving a considerable amount of work. Once all the reconfiguration scripts were generated it was possible to reconfigure the entire network. Given the relatively small amount of network nodes, the reconfiguration has been performed manually in each device in order to check the correct operation of the main network services:

- Internet access for the users and network devices
- RADIUS authentication of the users
- Local DNS cache
- Management access

The reconfiguration took about two months with just a small number of routers being reconfigured each day: this approach allowed us to gradually test the configuration and verify the correct functioning of the network as the number of reconfigured devices increased

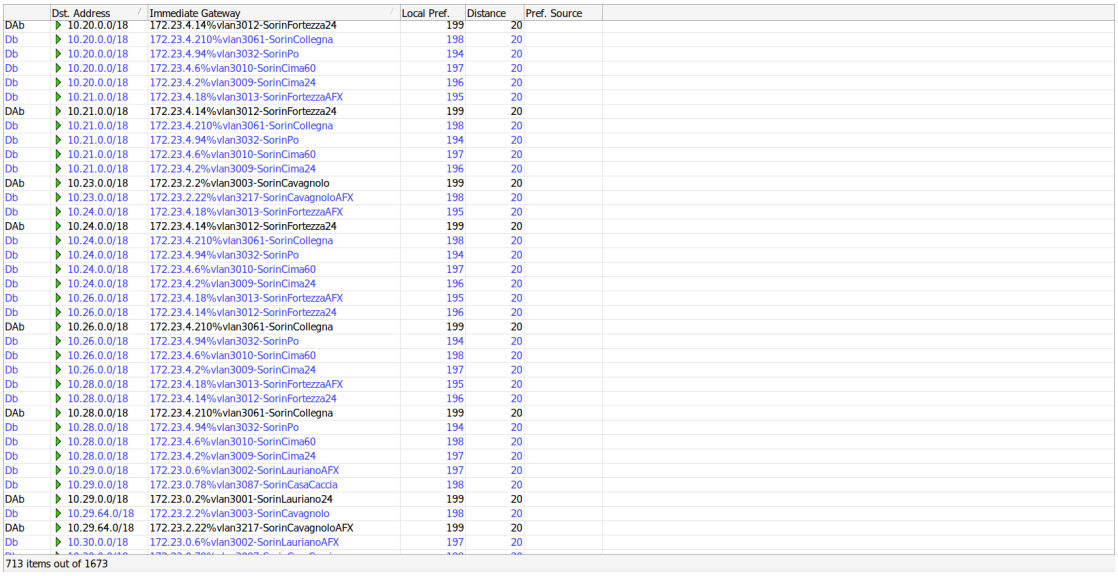
Chapter 8

Conclusions and future work

In this chapter, the behaviour of the network after the reconfiguration will be analyzed to prove its validity and functionality. In addition, some ideas and improvements that emerged during the development of this thesis project will be proposed.

8.1 Verification of the functionalities

After having configured all the network area that depends on the Sorin core router, which is one of the most complex area with a large number of loops, we can see that in the routing table of the core router multiple paths are available and the BGP local preference is correctly applied, with the highest one being preferred over the others. Hop by hop path can be manually configured by network operators



	Dst. Address	Immediate Gateway	Local Pref.	Distance	Pref. Source
DAb	10.20.0.0/18	172.23.4.14%vlan3012-SorinFortezza24	199	20	
Db	10.20.0.0/18	172.23.4.210%vlan3061-SorinCollegna	198	20	
Db	10.20.0.0/18	172.23.4.94%vlan3032-SorinPo	194	20	
Db	10.20.0.0/18	172.23.4.6%vlan3010-SorinCima60	197	20	
Db	10.20.0.0/18	172.23.4.2%vlan3009-SorinCima24	196	20	
Db	10.21.0.0/18	172.23.4.18%vlan3013-SorinFortezzaAFX	195	20	
DAb	10.21.0.0/18	172.23.4.14%vlan3012-SorinFortezza24	199	20	
Db	10.21.0.0/18	172.23.4.210%vlan3061-SorinCollegna	198	20	
Db	10.21.0.0/18	172.23.4.94%vlan3032-SorinPo	194	20	
Db	10.21.0.0/18	172.23.4.6%vlan3010-SorinCima60	197	20	
Db	10.21.0.0/18	172.23.4.2%vlan3009-SorinCima24	196	20	
DAb	10.23.0.0/18	172.23.2.2%vlan3003-SorinCavagnolo	199	20	
Db	10.23.0.0/18	172.23.2.22%vlan3217-SorinCavagnoloAFX	198	20	
Db	10.24.0.0/18	172.23.4.18%vlan3013-SorinFortezzaAFX	195	20	
DAb	10.24.0.0/18	172.23.4.14%vlan3012-SorinFortezza24	199	20	
Db	10.24.0.0/18	172.23.4.210%vlan3061-SorinCollegna	198	20	
Db	10.24.0.0/18	172.23.4.94%vlan3032-SorinPo	194	20	
Db	10.24.0.0/18	172.23.4.6%vlan3010-SorinCima60	197	20	
Db	10.24.0.0/18	172.23.4.2%vlan3009-SorinCima24	196	20	
Db	10.26.0.0/18	172.23.4.18%vlan3013-SorinFortezzaAFX	195	20	
Db	10.26.0.0/18	172.23.4.14%vlan3012-SorinFortezza24	196	20	
DAb	10.26.0.0/18	172.23.4.210%vlan3061-SorinCollegna	199	20	
Db	10.26.0.0/18	172.23.4.94%vlan3032-SorinPo	194	20	
Db	10.26.0.0/18	172.23.4.6%vlan3010-SorinCima60	198	20	
Db	10.26.0.0/18	172.23.4.2%vlan3009-SorinCima24	197	20	
Db	10.28.0.0/18	172.23.4.18%vlan3013-SorinFortezzaAFX	195	20	
Db	10.28.0.0/18	172.23.4.14%vlan3012-SorinFortezza24	196	20	
DAb	10.28.0.0/18	172.23.4.210%vlan3061-SorinCollegna	199	20	
Db	10.28.0.0/18	172.23.4.94%vlan3032-SorinPo	194	20	
Db	10.28.0.0/18	172.23.4.6%vlan3010-SorinCima60	198	20	
Db	10.28.0.0/18	172.23.4.2%vlan3009-SorinCima24	197	20	
Db	10.29.0.0/18	172.23.0.6%vlan3002-SorinLaurianoAFX	197	20	
Db	10.29.0.0/18	172.23.0.78%vlan3087-SorinCasaCaccia	198	20	
DAb	10.29.0.0/18	172.23.0.2%vlan3001-SorinLauriano24	199	20	
Db	10.29.64.0/18	172.23.2.2%vlan3003-SorinCavagnolo	198	20	
DAb	10.29.64.0/18	172.23.2.22%vlan3217-SorinCavagnoloAFX	199	20	
Db	10.30.0.0/18	172.23.0.6%vlan3002-SorinLaurianoAFX	197	20	

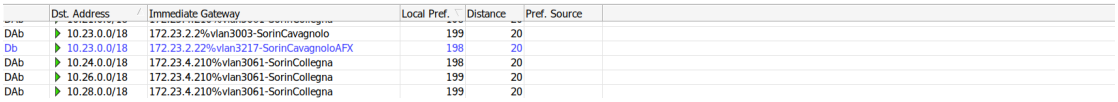
713 items out of 1673

Figure 8.1: Routing table of the Sorin core router after network reconfiguration

Considering the 10.24.0.0/18 route, the routes with the following next hops have a common upstream router in the *Rocca* node:

- 172.23.4.6
- 172.23.4.2
- 172.23.4.14
- 172.23.4.18

A failure is simulated by restarting that router: the 10.24.0.0/18 network is still reachable by means of the alternative paths that are not impacted by the fault. Traffic re-routing happened in less than 10 seconds without the intervention of network operators



	Dst. Address	Immediate Gateway	Local Pref.	Distance	Pref. Source
DAb	10.23.0.0/18	172.23.2.2%vlan3003-SorinCavagnolo	199	20	
Db	10.23.0.0/18	172.23.2.22%vlan3217-SorinCavagnoloAFX	198	20	
DAb	10.24.0.0/18	172.23.4.210%vlan3061-SorinCollegna	198	20	
DAb	10.26.0.0/18	172.23.4.210%vlan3061-SorinCollegna	199	20	
DAb	10.28.0.0/18	172.23.4.210%vlan3061-SorinCollegna	199	20	

Figure 8.2: Routing table of the Sorin core router when restarting a transport router

8.2 Performance evaluation

The main tasks performed by routers are CPU intensive, so CPU usage is the main parameter to consider when evaluating performance. A CPU utilization below 70% is considered safe and the router is operating correctly. In core routers, CPU usage is always below 50% even during peak hours so there are no bottlenecks

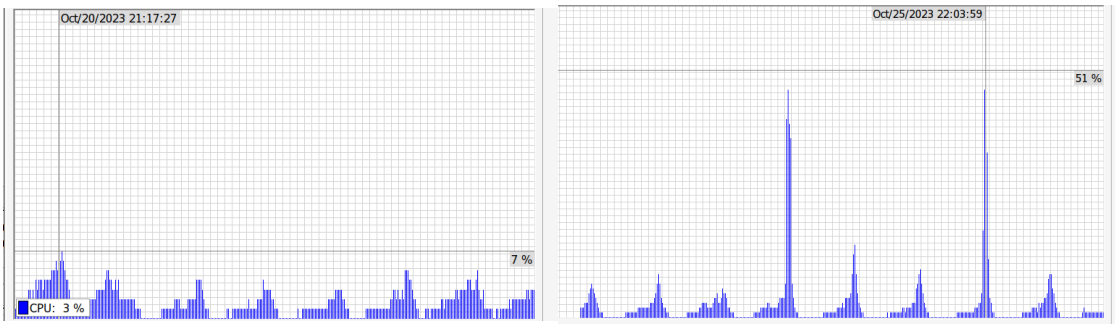


Figure 8.3: CPU usage graph over a week in a) datacenter router and b) Sorin router

In transit routers, thanks to L3 hardware offloading, the CPU usage is always below the limit. The following screenshots show resource utilization in one of the main transport routers of the network, with a throughput of 1,5 Gbit/s and 18 active BGP sessions

```
[SFSC@RB Switch Rocca] > /system/resource/print
                  uptime: 2d58m50s
                 version: 7.10 (stable)
              build-time: Jun/15/2023 05:17:29
        factory-software: 6.46.6
             free-memory: 164.0MiB
            total-memory: 256.0MiB
                     cpu: ARM
               cpu-count: 2
                cpu-load: 11%
          free-hdd-space: 1644.0KiB
         total-hdd-space: 16.0MiB
write-sect-since-reboot: 8819
       write-sect-total: 534331
      architecture-name: arm
             board-name: netPower 16P
               platform: MikroTik
```

```
[SFSC@RB Switch Rocca] > /system/resource/print
                  uptime: 2d59m13s
                 version: 7.10 (stable)
              build-time: Jun/15/2023 05:17:29
        factory-software: 6.46.6
             free-memory: 163.7MiB
            total-memory: 256.0MiB
                     cpu: ARM
               cpu-count: 2
                cpu-load: 93%
          free-hdd-space: 1644.0KiB
         total-hdd-space: 16.0MiB
write-sect-since-reboot: 8852
       write-sect-total: 534364
      architecture-name: arm
             board-name: netPower 16P
               platform: MikroTik
```

Figure 8.4: CPU usage in a transit router with a) L3 hardware offloading enabled and b) disabled

RAM usage is not a problem because it never exceeds 100 MB even with a large number of BGP sessions and all the routers are equipped with at least 128 MB of RAM

8.3 Scalability and management considerations

One of the requirements of the new configuration is that the management overhead must be as low as possible. One of the most complex and time-consuming operations is the moving of a node in the network by removing or adding links. The following operations must be performed:

- Identification of the new upstream and downstream nodes
- Definition of PTP VLANs and networks for the new links
- Configuration of the new upstream and downstream BGP peers
- Configuration of the address lists in case the network of the moved access point is announced in a new loop
- Removing addresses in the lists of old routers if needed
- Removing old BGP sessions, PTP VLANs and networks

A topology change that occurred after the network was reconfigured was the addition of a new link between the nodes in *Sgarbinato* and *Odalengo grande*, which originated a loop

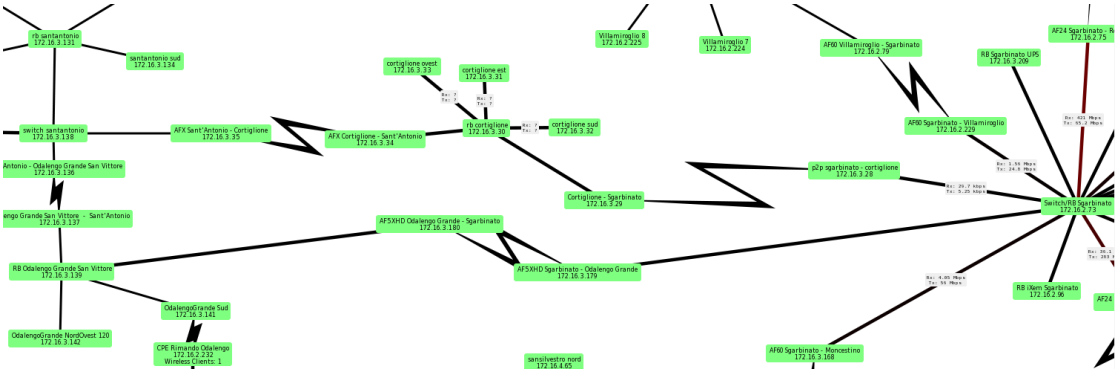


Figure 8.5: The new link between *Sgarbinato* and *Odalengo Grande*

The reconfiguration of the network section after the topology change took about 40 minutes, with most of the time spent configuring BGP sessions on the involved devices. Even if there are three loops between the area border gateway and the modified nodes, very few time (about 5 minutes) was required to adjust address lists in routers which are upstreams of a loop.

Another complex operation is the creation of a loop between two transit nodes carrying the traffic of a large number of access network. In this case, the most time-consuming operation is the configuration of address lists and priorities assignment

to links in the upstream router. When a loop was created between the *Rocca* and *Sgarbinato* nodes, the reconfiguration took about 30 minutes

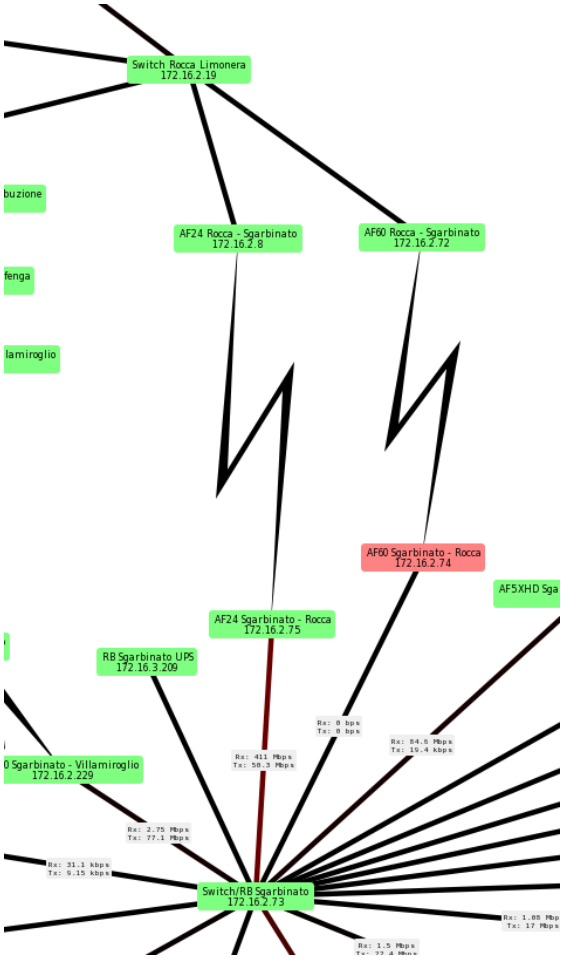


Figure 8.6: Adding the *AF60 Rocca - Sgarbinato* link created a loop between the nodes

In general the overhead caused by a topology change is quite low: only the directly involved routers must be reconfigured with the only exception being the routers in an upstream loop, if any.

8.4 Improvement opportunities

During the implementation of the new network configuration, the candidate identified some improvement areas, both in network design and implementation

- **Automatic configuration generation:** The new configuration is more complex with respect to the previous one. Configuring by hand all the devices in case of topology changes may lead to inconsistencies or errors in the configuration: it would be useful to have a software that, taking as input the information about the topology changes, produces as output the configuration scripts to be applied on the involved devices
- **Creation of an IP management network:** The management network is still based on VLANs which are statically filtered on the switch ports. If the link that carries the management VLAN fails, all the downstream devices are unreachable even if there are alternative paths available. By creating a dynamic IP management network it would be possible to exploit all the benefits of the new configuration
- **MPLS backbone:** Using the MPLS technology in the transport network would make it possible to make routing decisions taking into account the actual bandwidth of each link by using its traffic engineering capability. Unfortunately, MPLS label switching is done by the CPU but there are plans for implementing it in hardware [14]. The realization of an MPLS backbone is currently not feasible due to the highlighted limitations and it's outside of the scope of this thesis project

In general, the network configuration is not static: it will be improved in order to adapt to the network topology which is constantly evolving, possibly exploiting new functionalities that will be available in future RouterOS upgrades

Appendix A

Routers configuration script and constants

```
:local uplinkPorts { %UPLINKPORTS% }
:local uplinkVlanNames { %UPLINKVLANNAMES% }
:local uplinkIps { %UPLINKIPS% }
:local uplinkBgp { %UPLINKBGP% }
:local uplinkBgp6 { %UPLINKBGP6% }
:local ospfAreaId %OSPFAREAID%
:local ospfAreaPrefixLength %OSPFAREAPREFIXLENGTH%
:local routerName %ROUTERNAME%
:local downlinkPorts {%DOWNLINKPORTS%}
:local downlinkVlanNames {%DOWNLINKVLANNAMES%}
:local downlinkIPs {%DOWNLINKIPS%}
:local downlinkBgp {%DOWNLINKBGP%}
:local downlinkBgp6 {%DOWNLINKBGP6%}
:local as %AS%
:local apNetwork "%APNETWORK%"
:local hasDownlinks %HASDOWNLINKS%
:local ipv6Prefix %IPV6PREFIX%
:local loopbackIp ("%ROUTERID%")
:local routerId ("%ROUTERID%")
:local ipv6Network %IPV6NETWORK%

# Miscellaneous
/ip/route/set [find where dst-address="0.0.0.0/0"] disabled=yes
/ip/address set [find where address~$loopbackIp] disabled=yes
/interface/bridge add name="loopback"
/ip/address add address=$loopbackIp interface=loopback
/ipv6/settings/set disable-ipv6=no
/radius/set [find] src-address=$loopbackIp
```

```
29 /ppp/profile/set [find] dns-server=$routerId
30
31 # PTP VLANs and PTP IP networks
32 :foreach k,v in=$uplinkPorts do={
33   /interface/bridge/vlan/add bridge=bridge1 vlan-ids=$k tagged=( "
       bridge1," . "$v" )
34   /interface/vlan/add name=($uplinkVlanNames->$k) vlan-id=$k
       interface=bridge1
35   /ip/address/add address=($uplinkIps->$k) interface=(
       $uplinkVlanNames->$k)
36   /ipv6/address/add address=($ipv6Prefix.":f:".$k."::2/64") interface
       =($uplinkVlanNames->$k) advertise=no
37 }
38 :if ( $hasDownlinks ) do={
39   :foreach k,v in=$downlinkPorts do={
40     /interface/bridge/vlan add bridge=bridge1 vlan-ids=$k tagged=("
       bridge1," . $v)
41     /interface/vlan/add name=($downlinkVlanNames->$k) interface=
       bridge1 vlan-id=$k
42     /ip/address/add address=($downlinkIPs->$k) interface=(
       $downlinkVlanNames->$k)
43     /ipv6/address/add address=($ipv6Prefix.":2000:".$k."::1/64")
       interface=($downlinkVlanNames->$k) advertise=no
44   }
45 }
46
47 # Routing filters configuration and announced IP networks
48
49 /ipv6/route/add blackhole disabled=no dst-address=($ipv6Network)
       gateway=:: routing-table=main
50 /ipv6/firewall/address-list/add address=($ipv6Network) list=bgp-6
51 /ip/firewall/address-list/add list=bgp address=$apNetwork
52 /ip/route/add dst-address=$apNetwork blackhole
53 /routing/filter/rule add chain=bgp-out disabled=no rule="if (dst in
       bgp && dst-len==32) {reject}"
54 /routing/filter/rule/add chain=bgp-out disabled=no rule="accept;"
55 /routing/filter/rule/add chain=ospf-in disabled=no rule="if(dst-len
       == 128) {reject} else {accept}"
56
57 # OSPF configuration
58 /routing/ospf/instance/add name=$routerName originate-default=never
       routing-table=main router-id=$routerId
59 /routing/ospf/instance/add name=($routerName."-v6") version=3
       originate-default=never routing-table=main router-id=$routerId in-
       filter-chain=ospf-in
60
61 /routing/ospf/area/add area-id=$ospfAreaId instance=$routerName name=
       $ospfAreaId type=stub no-summaries
```

```
62 /routing/ospf/area/add area-id=$ospfAreaId instance=($routerName."-v6
       ") name=($ospfAreaId."-v6") type=stub no-summaries
63
64 /routing/ospf/interface-template add area=$ospfAreaId type=ptp
       networks=($ospfAreaId . $ospfAreaPrefixLength)
65 :local ospfInterfaces ""
66 :foreach name in=$uplinkVlanNames do={
67   :set ospfInterfaces ($ospfInterfaces.$name.",")
68 }
69 :foreach name in=$downlinkVlanNames do={
70   :set ospfInterfaces ($ospfInterfaces.$name.",")
71 }
72 /routing/ospf/interface-template add area=($ospfAreaId."-v6") type=
       ptp disabled=no interfaces="$ospfInterfaces"
73
74 # BGP configuration
75 /routing/bgp/template set default name=$routerName address-families=
       ip as=$as disabled=no hold-time=10s keepalive-time=2s output.
       default-originate=if-installed router-id=$routerId routing-table=
       main
76 :foreach name,ip in=$uplinkBgp do={
77   /routing bgp connection add address-families=ip as=$as disabled=no
       hold-time=10s keepalive-time=2s local.role=ebgp name=$name output.
       default-originate=if-installed .filter-chain=bgp-out .network=bgp
       remote.address=$ip router-id=$routerId routing-table=main
       templates=$routerName
78
79   /routing/bgp/connection add address-families=ipv6 as=$as disabled=
       no hold-time=10s keepalive-time=2s local.role=ebgp name=($name
       ."-6") output.default-originate=if-installed .filter-chain=bgp-out
        .network=bgp-6 remote.address=($uplinkBgp6->$name) router-id=
       $routerId routing-table=main templates=$routerName
80 }
81 :if ( $hasDownlinks ) do={
82   :foreach name,ip in=$downlinkBgp do={
83     /routing/bgp/connection add address-families=ip as=$as disabled=
       no hold-time=10s keepalive-time=2s local.role=ebgp name=$name
       output.default-originate=if-installed  remote.address=$ip router-
       id=$routerId routing-table=main templates=$routerName output.
       filter-chain=bgp-out .network=bgp
84
85     /routing/bgp/connection add address-families=ipv6 as=$as disabled
       =no hold-time=10s keepalive-time=2s local.role=ebgp name=($name
       ."-6") output.default-originate=if-installed .filter-chain=bgp-out
        .network=bgp-6 remote.address=($downlinkBgp6->$name) router-id=
       $routerId routing-table=main templates=$routerName
86   }
87 }
88
```

```
89 #FIREWALL
90 /ip/firewall/filter/add action=fasttrack-connection chain=forward
       connection-state=established,related
91 /ip/firewall/filter/add action=accept chain=forward connection-state=
       established,related
92 /ip/firewall/filter/add action=accept chain=input connection-state=
       established,related
93 /ip/firewall/filter/add action=drop chain=forward src-address
       =10.0.0.0/8 dst-address=172.16.0.0/16
94 /ip/firewall/filter/add action=drop chain=forward src-address
       =10.0.0.0/8 dst-address=172.23.0.0/16
95 /ip/firewall/filter/add action=drop chain=forward src-address
       =10.0.0.0/8 dst-address=10.0.0.0/8
96
97 /ipv6/firewall/filter add action=accept chain=input connection-state=
       established,related
98 /ipv6/firewall/filter add action=accept chain=input protocol=ospf
99 /ipv6/firewall/filter add action=accept chain=input protocol=tcp dst-
       port=179
100 /ipv6/firewall/filter add action=accept chain=input protocol=icmpv6
101 /ipv6/firewall/filter add action=drop chain=input dst-address
       =2000::/3
102 /ip/dns/set allow-remote-requests=yes
```

Variable name	**Type**	**Description**
uplinkPorts	Associative array	k is the PTP VLAN ID, v is the uplink ethernet interface
uplinkVlanNames	Associative array	k is the PTP VLAN ID, v is the name of the PTP VLAN
uplinkIps	Associative array	k is the PTP VLAN ID, v is the IP address of the uplink PTP network assigned to the router
uplinkBgp	Associative array	k is the name of the uplink BGP peer, v is its IP address in the PTP network
uplinkBgp6	Associative array	Information about IPv6 uplink BGP peers. The format is the same of uplinkBgp
ospfAreaId	String	The ID of the OSPF area
ospfAreaPrefixLength	String	The length of the IP prefix from which PTP networks are defined
routerName	String	The identity of the router
downlinkPorts, downlinkVlanNames, downlinkIPs, downlinkBgp, downlinkBgp6	Associative arrays	Information about BGP downstream peers and interfaces. The format is the same of the uplink variables
as	Integer	Private ASN of the router
apNetwork	String	Access IPv4 network of the router announced over IPv4 BGP sessions
hasDownlinks	Boolean	true if the router is a transit for other downstream routers, false if it's a terminal access node
ipv6Prefix	String	The IPv6 prefix of the network area
loopbackIp, routerId	String	The loopback IP of the router
ipv6Network	String	Access IPv6 network of the router announced over IPv6 BGP sessions

Appendix B

New devices configuration

```
#Datacenter router

#PTP
/interface vlan
add interface=sfp-sfpplus8 name=vlan4000-UtentiSorin vlan-id=4000
add address=172.16.0.1/30 interface=vlan4000-UtentiSorin network
    =172.16.0.0

#BGP
/routing bgp template
add address-families=ip as=207028 disabled=no hold-time=10s keepalive
    -time=2s name=itgate output.default-originate=if-installed router-
    id=172.16.0.1 routing-table=main
add address-families=ip as=207028 disabled=no hold-time=10s keepalive
    -time=2s local.role=ebgp name=sorinTorre output.default-originate=
    if-installed .filter-chain=bgp-out-downstream .redistribute=bgp
    remote.address=172.16.0.2/32 router-id=172.16.0.1 routing-table=
    main templates=itgate

#Routing filters
#Do not announce user networks and accept only PTP and user networks
    announcements
/routing filter rule
add chain=bgp-out-downstream disabled=no rule="if(dst in 10.0.0.0/8)
    {reject} else {accept}"
add chain=bgp-in-downstream disabled=no rule="if(dst in 10.0.0.0/8 ||
     dst in 172.23.0.0/16) {accept} else {reject}"
```

```
#Router Sorin
```

```
2
3 #PTP
4 /interface vlan
5 add interface=sfp-sfpplus1 name=vlan3001-SorinLauriano24 vlan-id=3001
6 add interface=sfp-sfpplus1 name=vlan3002-SorinLaurianoAFX vlan-id
    =3002
7 add interface=sfp-sfpplus1 name=vlan4000-UtentiSorin vlan-id=4000
8 /ip address
9 add address=172.23.0.1/30 interface=vlan3001-SorinLauriano24 network
    =172.23.0.0
10 add address=172.23.0.5/30 interface=vlan3002-SorinLaurianoAFX network
    =172.23.0.4
11 add address=172.16.0.2/30 interface=vlan4000-UtentiSorin network
    =172.16.0.0
12
13 #OSPF configuration
14 /routing ospf instance
15 add disabled=no name=ospv-v2 redistribute="" router-id=172.16.7.109
    routing-table=main
16 add disabled=no in-filter-chain=ospf-in name=ospf-v3 originate-
    default=never router-id=172.16.0.96 version=3
17 /routing ospf area
18 add area-id=172.23.0.0 disabled=no instance=ospv-v2 name=lauriano no-
    summaries type=stub
19 /routing ospf interface-template
20 add area=lauriano disabled=no networks=172.23.0.0/23 type=ptp
21
22 #BGP configuration
23 /routing bgp template
24 add address-families=ip as=2096 disabled=no hold-time=10s keepalive-
    time=2s name=downstream output.default-originate=if-installed .
    filter-chain=bgp-out-nousersnet router-id=172.16.7.109 routing-
    table=main
25 /routing bgp connection
26 add address-families=ip as=2096 disabled=no hold-time=10s input.
    filter=bgp-in-lauriano24 keepalive-time=2s local.role=ebgp name=
    lauriano24 output.default-originate=if-installed .filter-chain=bgp
    -out-nousersnet remote.address=172.23.0.2/32 router-id
    =172.16.7.109 routing-table=main templates=downstream
27 add address-families=ip as=2096 disabled=no hold-time=10s input.
    filter=bgp-in-laurianoAFX keepalive-time=2s local.role=ebgp name=
    laurianoAFX output.default-originate=if-installed .filter-chain=
    bgp-out-nousersnet remote.address=172.23.0.6/32 router-id
    =172.16.7.109 routing-table=main templates=downstream
28
29 #Address lists
30 /ip firewall address-list
31 add address=10.180.0.0/18 list=1-lauriano24
32 add address=10.180.0.0/18 list=2-laurianoAFX
```

```
add address=10.35.0.0/18 list=1-laurianoAFX
add address=10.35.0.0/18 list=2-lauriano24

#Routing filters
/routing filter rule
add chain=bgp-in-lauriano24 disabled=no rule="if(dst in 1-lauriano24)
    {set bgp-local-pref 199; accept;}"
add chain=bgp-in-lauriano24 disabled=no rule="if(dst in 2-lauriano24)
    {set bgp-local-pref 198; accept;}"
add chain=bgp-in-lauriano24 disabled=no rule="set bgp-local-pref 99;
    accept;"
add chain=bgp-in-laurianoAFX disabled=no rule="if(dst in 1-
    laurianoAFX) {set bgp-local-pref 199; accept;}"
add chain=bgp-in-laurianoAFX disabled=no rule="if(dst in 2-
    laurianoAFX) {set bgp-local-pref 198; accept;}"
add chain=bgp-in-laurianoAFX disabled=no rule="set bgp-local-pref 98;
    accept;"
```

```
#Router Lauriano scuola

#Loopback configuration
/interface bridge add
name=loopback
/ip address
add address=10.180.0.2 interface=loopback

#PTP
/interface vlan
add interface=bridge1 name=vlan3001-SorinLauriano24 vlan-id=3001
add interface=bridge1 name=vlan3002-SorinLaurianoAFX vlan-id=3002
add interface=bridge1 name=vlan3008-LaurianoMoriondo vlan-id=3008
/ip address
add address=172.23.0.2/30 interface=vlan3001-SorinLauriano24 network
    =172.23.0.0
add address=172.23.0.6/30 interface=vlan3002-SorinLaurianoAFX network
    =172.23.0.4
add address=172.23.0.9/30 interface=vlan3008-LaurianoMoriondo network
    =172.23.0.8

#OSPF
/routing ospf instance
add disabled=no name=LaurianoScuola originate-default=never router-id
    =10.180.0.2 routing-table=main
add disabled=no in-filter-chain=ospf-in name=RBSwitchLaurianoScuola-
    v3 originate-default=never router-id=10.180.0.2 routing-table=main
     version=3
/routing ospf area
```

```
24 add area-id=172.23.0.0 disabled=no instance=LaurianoScuola name
       =172.23.0.0 no-summaries type=stub
25 add area-id=172.23.0.0 disabled=no instance=RBSwitchLaurianoScuola-v3
        name=172.23.0.0-v3 no-summaries type=stub
26 /routing ospf interface-template
27 add area=172.23.0.0 disabled=no networks=172.23.0.0/23 type=ptp
28 add area=172.23.0.0-v3 disabled=no interfaces="vlan3001-
       SorinLauriano24,vlan3002-SorinLaurianoAFX,vlan3008-
       LaurianoMoriondo,vlan3071-LaurianoMezzana,vlan3072-
       LaurianoCorneliana,vlan3073-LaurianoScuolaCampanile,vlan3075-
       LaurianoScuolaPiazzoCimiteroCampanile" type=ptp
29
30 #BGP
31 /routing bgp template
32 set default address-families=ip as=180 disabled=no hold-time=10s
       keepalive-time=2s output.default-originate=if-installed .network=
       bgp-networks router-id=10.180.0.2 routing-table=main
33 /routing bgp connection
34 add address-families=ip as=180 disabled=no hold-time=10s input.filter
       =bgp-in-default1 keepalive-time=2s local.role=ebgp name=
       SorinLauriano24 output.default-originate=if-installed .filter-
       chain=bgp-out .network=bgp remote.address=172.23.0.1/32 router-id
       =10.180.0.2 routing-table=main templates=default
35 add address-families=ip as=180 disabled=no hold-time=10s input.filter
       =bgp-in-default2 keepalive-time=2s local.role=ebgp name=
       SorinLaurianoAFX output.default-originate=if-installed .filter-
       chain=bgp-out .network=bgp remote.address=172.23.0.5/32 router-id
       =10.180.0.2 routing-table=main templates=default
36 add address-families=ip as=180 disabled=no hold-time=10s keepalive-
       time=2s local.role=ebgp name=LaurianoScuolaMoriondo output.default
       -originate=if-installed .filter-chain=bgp-out .network=bgp remote.
       address=172.23.0.10/32 router-id=10.180.0.2 routing-table=main
       templates=default
37
38 #Routing filters
39 /routing filter rule
40 add chain=bgp-out disabled=no rule="if (dst in bgp && dst-len==32) {
       reject}"
41 add chain=bgp-out disabled=no rule="accept;"
42 add chain=bgp-in-default1 disabled=no rule="if(dst == 0.0.0.0/0) {set
        bgp-local-pref 199; accept;} else {accept}"
43 add chain=bgp-in-default2 disabled=no rule="if(dst == 0.0.0.0/0) {set
        bgp-local-pref 198; accept;} else {accept}"
44 add chain=ospf-in disabled=no rule="if(dst-len == 128) {reject} else
       {accept}"
45
46 #L3 hardware offloading
47 /interface ethernet switch set 0 l3-hw-offloading=yes
48 /interface ethernet switch port set [find] l3-hw-offloading=yes
```

```
#RB Moriondo 1
#Loopback
/interface bridge
add name=loopback
/ip address add
address=10.35.0.2 interface=loopback

#PTP
/interface vlan
add interface=ether1 name=vlan3008-LaurianoMoriondo vlan-id=3008
/ip address
add address=172.23.0.10/30 interface=vlan3008-LaurianoMoriondo
    network=172.23.0.8

#OSPF
/routing ospf instance
add disabled=no name=moriondo router-id=10.35.0.2 routing-table=main
add disabled=no in-filter-chain=ospf-in name=moriondo-v3 originate-
    default=never router-id=10.35.0.2 routing-table=main version=3
/routing ospf area
add area-id=172.23.0.0 disabled=no instance=moriondo name=lauriano no
    -summaries type=stub
add area-id=172.23.0.0 disabled=no instance=moriondo-v3 name=lauriano
    -v3 no-summaries type=stub
/routing ospf interface-template
add area=lauriano disabled=no networks=172.23.0.0/23 type=ptp
add area=lauriano-v3 disabled=no interfaces=vlan3008-LaurianoMoriondo
     type=pt

#BGP
/routing bgp template
add address-families=ip as=35 disabled=no hold-time=10s keepalive-
    time=2s name=default output.filter-chain=bgp router-id=10.35.0.2
    routing-table=main
/routing bgp connection
add address-families=ip as=35 disabled=no hold-time=10s keepalive-
    time=2s local.address=172.23.0.10 .role=ebgp name=lauriano output.
    filter-chain=bgp-out .network=bgp remote.address=172.23.0.9/32
    router-id=10.35.0.2 routing-table=main templates=default

#Routing filters
/routing filter rule
add chain=bgp-out disabled=no rule="if (dst in bgp && dst-len == 32)
    {reject;}"
add chain=bgp-out disabled=no rule="accept;"
add chain=ospf-in disabled=no rule="if(dst-len == 128) {reject} else
    {accept}"
```

Bibliography

[1] Senza Fili Senza Confini. *Chi siamo*. URL: https://senzafilisenzaconfini.org/AboutUS.php (cit. on p. 1).

[2] Wikipedia. *OSI model*. URL: https://en.wikipedia.org/wiki/OSI_model (cit. on p. 4).

[3] Cisco. *Open Shortest Path First*. URL: https://www.cisco.com/c/it_it/support/docs/ip/open-shortest-path-first-ospf/7039-1.html (cit. on p. 12).

[4] Cisco. *Border Gateway Protocol*. URL: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-16/irg-xe-16-book/configuring-a-basic-bgp-network.html (cit. on p. 14).

[5] Eric Rosen, Arun Viswanathan, and Ross Callon. *RFC3031: Multiprotocol label switching architecture*. 2001 (cit. on p. 15).

[6] Geo Sat Tech. *Frequency attenuation in the atmosphere*. URL: https://propagation.ece.gatech.edu/ECE6390/project/Fall2012/Team09/Team9GeoSatTech_website_FINAL/SatCom%20website/images/GaseousAttenuation.png (cit. on p. 21).

[7] Mikrotik. *The RouterOS operating system*. URL: https://help.mikrotik.com/docs/display/ROS/RouterOS (cit. on p. 24).

[8] Teldat. *Connecting some of the dots between Nftables, Iptables and Netfilter*. URL: https://www.teldat.com/blog/nftables-and-netfilter-hooks-via-linux-kernel/ (cit. on p. 26).

[9] Mikrotik. *Ethernet routers*. URL: https://mikrotik.com/products/group/ethernet-routers (cit. on p. 28).

[10] Wikipedia. *Multiprotocol label switching*. URL: https://it.wikipedia.org/wiki/Multiprotocol_Label_Switching#MPLS_Traffic_Engineering (cit. on p. 49).

[11] Mikrotik. *Mikrotik L3 hardware offloading*. URL: https://help.mikrotik.com/docs/display/ROS/L3+Hardware+Offloading (cit. on pp. 49, 59).

[12] Mikrotik. *Mikrotik CRS318-16P-2S+ block diagram*. URL: https://i.mt.lv/cdn/product_files/CRS318-16P-2S_220938.png (cit. on p. 49).

[13] APNIC. *IPv6 address planning guidelines*. URL: https://blog.apnic.net/2019/08/22/how-to-ipv6-address-planning/ (cit. on p. 58).

[14] Mikrotik. *MPLS hardware offloading in Mikrotik devices*. URL: https://forum.mikrotik.com/viewtopic.php?t=181017 (cit. on p. 72).

Acknowledgements

Giunto al termine di questo lavoro, ci tengo particolarmente a ringraziare il mio relatore e presidente dell'associazione Daniele Trinchero per la grande opportunità che mi ha concesso, assegnandomi il ruolo di direttore tecnico della rete. Fin dal primo giorno di assunzione, la fiducia completa che ha riposto nel mio operato mi ha permesso di crescere moltissimo professionalmente, sia lato tecnico che umano: una delle abilità più importanti che ho imparato in associazione è sicuramente la capacità di fare squadra, unendo le forze per raggiungere i risultati prefissati. A tal proposito, ringrazio anche tutti i miei colleghi in Associazione e negli iXem Labs per la guida e per tutto il lavoro che è stato svolto sulla rete, il quale ha reso possibile la realizzazione di questo progetto.

Vorrei anche utilizzare questo spazio finale per scrivere due righe a tutte quelle persone che in qualche modo sono entrate nella mia vita e mi sono state vicine, sia durante la stesura di questo elaborato che durante il mio percorso universitario.

A mia madre, che ha sempre mostrato il suo sostegno nei miei confronti e non si è mai tirata indietro nei momenti di bisogno.

A mio padre, la cui grande professionalità, dedizione e precisione nel lavoro è stata fonte di ispirazione durante il mio percorso universitario.

Ai miei nonni, che hanno sempre fatto il tifo per me fin dal primo giorno.

A tutti i miei amici, sia Torinesi che non, e ai coinquilini di via Felizzano, che hanno camminato al mio fianco per una parte o per tutta la durata di questo percorso e hanno condiviso con me gioie e dolori.