Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# MASTER THESIS

**Title:** Detecting Network issues in Wireless Community Networks

**Master Degree:** Master in Science in Telecommunication Engineering & Management

**Author:** Alberto Andrés Rodríguez

**Director:** Roc Meseguer Pallarès

**Date:** March 13th 2013

**Title:** Detecting issues in Wireless Community Networks

**Author:** Alberto Andrés Rodríguez

**Director:** Roc Meseguer Pallarès

**Date:** March 13th 2013

## Overview

The possibilities that wireless technology offers have facilitated that multiple Internet communities have been built, independent of new business models on networks. These network communities had the aim to connect users in order to be able to share resources and services. Such communities are those that have established community networks like Guifi.net. Guifi.net is also a space for research and development of technologies for adaptation to the specific WiFi in an open network.

The principal objectives of this master thesis are to study the different issues that could be found in a wireless community network such as Guifi.net. These problems are commonly due of mixing routing protocols. For this reason the scenarios will be reproduced it by a software network emulator, then with the data collected will be treated to give a solution to these problems. Furthermore, this thesis has a special interest in the protocols used in wireless community networks and how these routing protocols connect the different nodes of the network. Also it will take into account the behavior of the network when a node is added or deleted and how this action can have a large impact to the entire network.

It is an interesting research of this kind of communities because nowadays people are becoming aware that communities' networks are made to allow people to connect with people around the world and this master thesis will help to understand the problems that would appear in a wireless community and give a solution to these problems.

For the development of the tasks listed above it was necessary to monitor Guifi.net network for a determined period of time in order to detect current issues of the network and understand how the networks works, as well as the study of other concepts. To carry out the execution of this master thesis has been used Netkit network emulation software.

**Títol:** Detectar problemes a les comunitats de xarxes sense fils

**Autor:** Alberto Andrés Rodríguez

**Director:** Roc Meseguer Pallarès

**Data:** 13 de Març de 2013

## Resum

Les possibilitats que ofereix la tecnologia sense fils han facilitat que múltiples comunitats d'Internet s'hagin construït, amb independència de nous models de negoci en les xarxes. Aquestes comunitats de xarxa tenen l'objectiu de connectar als usuaris amb la finalitat de ser capaços de compartir recursos i serveis. Aquestes comunitats són les que han establert xarxes comunitàries com Guifi.net. Guifi.net és també un espai per a la investigació i desenvolupament de tecnologies per a l'adaptació a l'específica Wi-Fi a una xarxa oberta.

Els objectius principals d'aquest projecte són l'estudi dels diferents problemes que es poden trobar en una xarxa sense fils comunitària com pot ser Guifi.net. Aquests problemes son comunment causats al barrejat diferents protocols d'enrutament per aquesta raó es reproduïran els escenaris mitjaçant un emulador de xarxes i amb les dades recollides es tractará de proporcionar una posible solució a aquests problemes. A més, aquest projecte té un interès especial en els protocols utilitzats en les xarxes sense fils comunitàries i dels protocols d'enrutament que connecten els diferents nodes de la xarxa. També es té en compte el comportament de la xarxa quan un node s'afegeix o s'elimina i com aquesta acció pot tenir un gran impacte a tota la xarxa.

Aquesta tesi l'ajudarà a entendre i detectar els problemes que pugin apareixer a una comunitat wireless i dona'ls-hi una solució.

Per al desenvolupament de les tasques esmentades anteriorment ha sigut necessari monitoritzar la xarxa Guifi.net durant un determinat període de temps per tal de detectar problemes actuals que sofreix la xarxa i al mateix entendre com funciona, així com l'estudi d'altres conceptes. Per dur a terme l'execució d'aquest projecte s'ha utilitzat el programari de emulació de xarxes Netkit.

# INDEX

# INTRODUCTION

Currently wireless infrastructure can be built at a very low cost compared to traditional wired alternatives. However building a wireless network is not only about saving money, but building a wireless network will provide to a community of users the opportunity to access to the information easier and cheaper. Wireless communities networks are connected to Internet at high speeds and are involved in the global market. People around the world are finding that the access to Internet gives them an opportunity to discuss their problems, policies or anything that could be important in their lives. But even without Internet access, community wireless networks have a great value. Allow people to collaborate on projects over long distances. Voice communications, electronic mail and other data can be exchanged at low cost. By involving community members in the construction of the network, knowledge and trust can be spread throughout the community and people could begin to understand the importance of taking part in such communities. Nowadays people are becoming aware that communities' networks are made to allow people to connect with people around the world.

The main objectives of this thesis are to assess, describe and try to automatically detect topological and routing issues that a wireless community network such as Guifi.net could have. During the execution of this master thesis will be evaluated how the network is distributed. This information will be used to known the different routing protocols that actually the network uses. Once known which ones are most used, these routing protocols will be emulated under different scenarios in order to reproduce their behavior in the network. This work will be used to reproduce the issues located in the network and to search a possible solution to them. Due to the impossibility of using the Guifi.net network as a scenario to perform the tests, because several users and companies depends their connection from this network, it has been decided to use software called Netkit. This software allows emulating virtual networks devices with the same features of the systems that can be found in real production environments, including configuration, and communication protocols, leaving in the background network performance.

Chapter 1 provides an introduction to the Wireless Community Networks, specifically about the community Guifi.net where the master thesis have been developed. It describes the purpose of the community and current technologies used by the community. It describes the hardware and software used in the network. In this chapter it will also be explained with more detail the different useful routing protocols for this kind of networks.

In chapter 2 will be explained how the network is structured and the main problems that could appear in a wireless community network as Guifi.net. Also it will be explained how the routing protocols are used in this network and the problems that originate when are mixed. This chapter is the first part of the development, which will help to understand how the packets are transmitted in a network and how this problem affects to the traffic of the network and the functioning of it.

The network software emulation used for this master thesis will be introduced in Chapter 3.

Chapter 4 relates to the second part of the development of this study. In this chapter are showed the tests done in the described scenarios, reproducing the same problems that could appear in real networks when BGP and OSPF routing protocols are used in the same network. These protocols will be implemented in different scenarios, each referring to the issues existing in Guifi.net. Each scenario will set up to study and verify the operation of the protocol features. The information obtained by setting up the labs will be very useful to examine possible solutions to these problems or try to introduce some improvements.

Possible solutions that could be applied to solve the problems stated in Chapter before are stated in Chapter 5. This Chapter tries to recollect all the information of previous chapters to apply it in the network in order to avoid the problems that the network actually has.

In Chapter 6 propose possible solutions to apply to the problems studied in the particular case of Guifi.net. In this proposal will be applied the knowledge obtained in the previous chapters, in order to achieve a network with less problems than it could has now. This will be obtained by the application of different kind of filters that would optimize the traffic that are in the network.

Finally the conclusions of this master thesis are stated in Chapter 7. It is based on the emulations performed and the recollected data during the realization of this master thesis. It also includes future lines of work.

The annexes cover the whole range of technical information when implementing the different case studies of practical chapters.

# CHAPTER 1. WIRELESS COMMUNITY NETWORKS

## 1.1 What is a Wireless Community Network?

The technological developments of the last few years have enabled new forms of interactions and collaboration between individuals. Wireless community networks consist on a group of volunteers, institutions and companies who build open and free networks, these networks provides access to the Internet over the airwaves, being an alternative to private ISP's networks. This kind of networks uses two way radios to take advantage of the unregulated portion of the telecommunications spectrum. Wireless community networks are organized geographically to serve a specific neighborhood or community and to improve the quality of life within that community.

These networks are characterized by the linkage of many individuals in a complex web, offering flexibility and adaptability when something goes wrong. Community wireless networks have the potential to provide affordable Internet accessibility and democratically controlled. Also, these kinds of networks are open and free, any person or administration could see how the network is structured and these networks do not impose restrictions. In addition, these kinds of networks offer a new tool for community economic development that can reduce poverty, promote a sense of community, and encourage civic participation.

## 1.2 Case of study: Guifi.net

Guifi.net[1] was born in 2004, is a nonprofit telecommunications community network based in an interconnection agreement where each participant set up nodes and connects them to the network in order to extend it while gaining connectivity. Guifi.net participants are individuals, companies and administrations that constitute a public telecommunications network. Currently have over 17.000 actives nodes and 32.000 kilometers of wireless links.

Guifi.net has declared as an open, free and neutral network for the following reasons:

- It has been considered as open because network data configuration is published, any person or administration could see how the network is structured and therefore has the option to improve it or extended it.

- It has been considered as free because does not impose restrictions.

- It has been considered as neutral because in the network can be found any content that someone needs.

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

Most of the links used in Guifi.net are wireless links, for this purpose are used the available frequencies, but at the same time there are sections of network made by fiber optic connections. The methods of connection to the network could be simple connections or connections that extend the network. The simple connections are used to connect to other point of the network with the same conditions. In case of extended connections, are used to extend the network with the same conditions as the original link.

The joining to the network could be individual or collectively and at any time members can renounce to their membership. The terms of the membership are described in the project website, although these terms basically says that Guifi.net is not responsible for the content of the network and that anyone is free to use the network for any purpose.

These wireless community networks use the spectrum of 5 GHz, this spectrum is used because allows transmit information without paying any license.

Other interesting data about Guifi.net is their annual growth, over 2.500 new users every year. This growth represents more than 2.500 Petabyte of network transfer information.

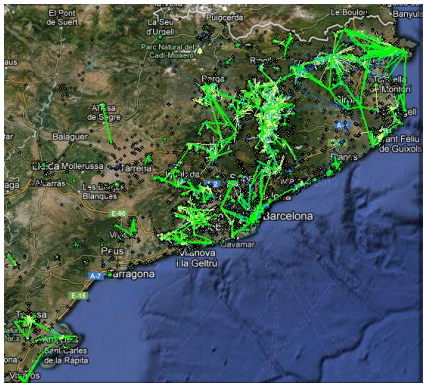



***Figure 1.1:*** *Guifi.net network in Catalonia*

## 1.3 Technologies used in Guifi.net

Once explained the social environment of Guifi.net, is it necessary to observe how the networks is technologically developed. This point is necessary in order to know what tools are available to perform this study and show in which technologies we want to focus on. Currently, Guifi.net as most of wireless community networks has several topologies in use, but as it was explained before everyone is free to use whatever hardware and software wants, as long as it fits in Guifi.net philosophy. But as in most networks, networks technologies can be separated in software and hardware.

### 1.3.1 Hardware

The network Guifi.net is based basically on backbone hardware and in end users hardware. But the highly meshed topology implies that much of the infrastructure is the same users' computers. Thus, the network consists of a large amount of access points.

#### *1.3.1.1 End users hardware*

Currently, in Guifi.net network end users can use any low cost WiFi router, but the current most used hardware is Ubiquity NanoStation[2], which is a low cost CPE, featuring an easy-to-use firmware, and with a polite design suitable for everyone. Before that, Linksys WRT54GL was extensively used.

***Figure 1.2:*** *Ubiquity NanoStation*

#### *1.3.1.2 Network backbone hardware*

Like as end users case, on the backbone network of Guifi.net, the most used hardware is Mikrotik's RouterBoards[3], which features a great performance and an affordable price. But is it a possible example of hardware. Other CWN use other hardware for the backbone or for end user.

***Figure 1.3:*** *Mikrotik RouterBoard*

Commonly, in Guifi.net end users use sector antennas but for the network backbone the most used type is panel antennas. In other Wireless Community Networks other hardware could be used, it depends on the features and the knowledge of the users.

### 1.3.2 Software

Usually, the software used for the end user and backbone hardware is the manufacturer's software. But it is important to choose those distributions that offer more freedom of operation and availability. Therefore, a very good alternative are OpenWRT[4] and DD-WRT[5], an open source firmware's fully customizable which supports plenty of hardware

## 1.4 Routing protocols used in Guifi.net

Routing protocols are used to facilitate the exchange of routing information between routers, also allow routers to dynamically learn information about remote networks and automatically add this information to their own routing tables.

[2] http://www.ubnt.com/products/nano.php
[3] http://www.routerboard.com
[4] http://www.openwrt.org
[5] http://www.dd-wrt.org

Routing protocols determine the best path to each network, which it is added to the routing table. The benefits of using a dynamic routing protocol is routers are likely to exchange routing information whenever there is a topology change. This exchange allows routers to automatically learn about new networks and also to find alternate paths if there is a link failure to a current network.

The protocols that currently Guifi.net uses can be classified in two main groups:

- Dynamic protocols: are used by routers to automatically learn about remote networks from other routers. This kind of protocols can be classified according to various characteristics, but the most commonly used routing protocols are as follows: BGP, OSPF and RIP. The main operations of this kind of protocols are the next:

  - The router sends and receives routing messages on its interfaces.
  - The router shares routing messages and routing information with other routers that are using the same routing protocol.
  - Routers exchange routing information to learn about remote networks.
  - When a router detects a topology change, the routing protocol can advertise this change to other routers

- Routing Protocols optimized for mobile ad-hoc networks: This kind of protocols uses hello messages to discover new node and then disseminate information throughout the mobile ad-hoc network. The most used protocols for ad-hoc networks are the protocols OLSR, BMX6, BATMAN.

Currently the 90% of the routing protocols used in the network are BGP and OSPF and the routing protocols for mobile ad-hoc networks only are used in small communities. According to that, this master thesis will be focused only in BGP and OSPF routing protocols.

The following sections are intended to explain routing protocols to those people which are not familiarized with networking.

### 1.4.1 Definition of AS

According to the classic definition of an Autonomous System, is a collection of routers under a common administration. An Autonomous System is a unique network identification to communicate with the others Autonomous Systems of the network. Due of the Internet is based on the Autonomous System concept; two types of routing protocols are required: interior and exterior routing protocols. These protocols are:

### 1.4.2 IGP and EGP

- Interior gateway protocols (IGP): Used for intra-autonomous system routing, that is, routing inside an Autonomous System.
- Exterior gateway protocols (EGP): Used for inter-autonomous system routing, that is, routing between Autonomous Systems.

In one hand, corporate networks, such as academic or business normally uses IGP, to route within the Autonomous System and also used to route within the individual networks themselves. In the other hand, EGP's are designed for use between different autonomous systems that are under the control of different administrations.

### 1.4.3 BGP

The Border Gateway Protocol is a routing protocol between autonomous systems. It was defined in order to solve certain problems and limitations of its predecessor, the Exterior Gateway Protocol (EGP).

The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems to know the path to other networks. This network reachability information includes information on the list of Autonomous Systems (AS's in advance) that reachability information traverses. This list is useful to differentiate the connectivity between different AS's in order to avoid routing loops and establish policies that concerns AS's. BGP uses TCP as its transport protocol and is virtually present in all commercial routers and hosts.

The BGP is a routing protocol very robust and scalable, as evidenced by the fact that it is the routing protocol most used on the Internet. Currently, BGP routing tables have more than 10.000 routes. To achieve scalability at this level, BGP uses route parameters, called attributes, to define routing policies and maintain a stable environment.

When BGP is used between different AS's, is called external BGP (eBGP). If a service provider uses to exchange routes within the same autonomous system, then is called internal BGP (iBGP).

Currently in Guifi.net community network there are more than 2.600 links made by BGP protocol. The Figure shows the BGP links in the region of Catalonia, the red points shown in the picture are the BGP links and nodes that currently are in the network.

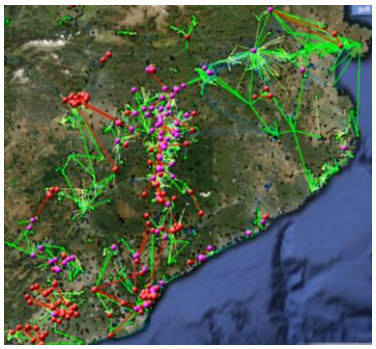


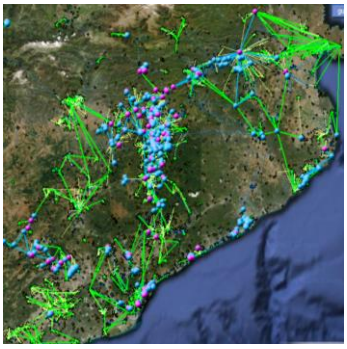
***Figure 1.4:*** *Guifi.net BGP links*

### 1.4.4 OSPF

OSPF is a routing protocol used to determine the best route for delivering the packets within an IP (Internet Protocol) networks. OSPF is perhaps the most widely used Interior Gateway Protocol (IGP) in large enterprise networks

The OSPF protocol is based on a link-state algorithm. The state link is passed through LSA's (Link State Advertisement) from available routers and constructs a topology map of the network. Routers exchange this information in order to make a global database of the network. With this database, is it possible to obtain a complete map of the network and compute optimal routes to each destination using Dijkstra's[6] algorithm. An OSPF network may be structured, or subdivided, into routing areas to simplify administration and optimize traffic and resource utilization.

When OSPF routing protocol detects changes in the topology, such as link failures, it converges very quickly on a new loop-free routing structure within seconds. It computes the shortest path tree for each route using the Dijkstra's algorithm.



Currently OSPF has more than 3.300 links in the Guifi.net network, the figure show all these links, the blue points shown in the Figure, represents the OSPF nodes and links that currently are in the network.

***Figure 1.5:*** *Guifi.net OSPF links*

[6] http://www.dgp.toronto.edu/people/JamesStewart/270/9798s/Laffra/DijkstraApplet.html

### 1.4.5 Summary of Routing Protocols

In the next lines will summarize the main features of these protocols.

OSPF is an IGP, while BGP is an EGP; IGP should not be confused with iBGP.

BGP is used at the edge of the network to connect the network to the Internet, but OSPF is used internally inside a network.

OSPF employs a hierarchical network design using areas; also OSPF will form neighbor relationships with adjacent routers in the same Area.

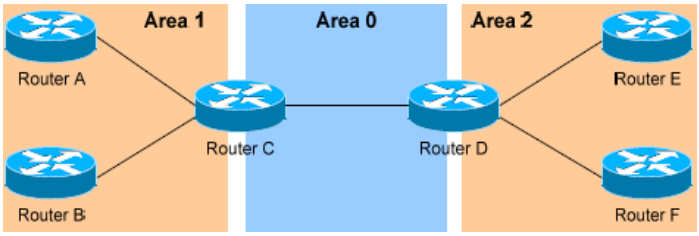



***Figure 1.6:*** *OSPF areas*

There are two types of BGP neighbor relationships:

- iBGP: BGP neighbors within the same autonomous system.
- eBGP: BGP neighbors connecting separate autonomous systems.

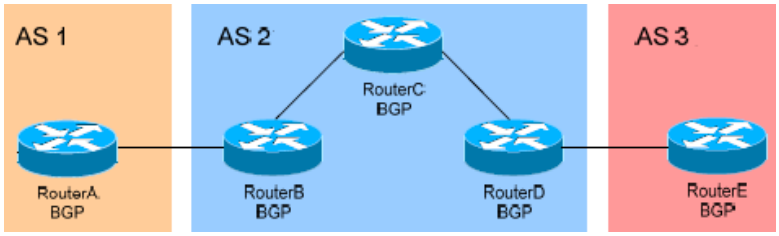



***Figure 1.7:*** *iBGP and eBGP links*

In the above figure, RouterB and RouterC in AS2 would form an iBGP link, while RouterA in AS1 and RouterB in AS2 would form an eBGP link.

BGP uses TCP as transport protocol while OSPF uses IP.

The next table summarizes the main features of the BGP and OSPF protocols.

***Table 1.1:*** *Features of BGP and OSPF routing protocols*

| **Features** | **BGP** | **OSPF** |
|---|---|---|
| Protocol type | Path Distance Vector | Link state |
| Classless | Yes | Yes |
| Variable Length Subnet Masking | Yes | Yes |
| Automatic Summarization | Yes | No |
| Size | Very Large | Large |
| Convergent Time | Slow | Fast |
| Metric | Path Attribute | Cost |
| Complexity to configure | High | Medium |
| Boundary Separation | Autonomous Systems | Areas |
| Administrative internal distance | 200 | 110 |
| Administrative external distance | 20 | N/a |

## 1.5 Future of the network

Over the past few years, Guifi.net network has been growing at a huge rate. Each year the number of nodes is twice than the previous year, so a year by now is it able to speak of over 17.000 nodes. The future challenges for the community are to find an even more auto configurable method of connecting nodes to the network, minimizing the work to be done by the end user, and the necessary skills to do it. Within this way it will be possible to avoid the problems explained in this master thesis.

# CHAPTER 2. MAIN PROBLEMS IN THE NETWORK GUIFI.NET

## 2.1 Current configuration in Guifi.net

The current network management areas in Guifi.net are for countries, regions, cities, neighborhoods, etc. but currently there are no technical areas of the network, i.e. areas OSPF, BGP. For that reason is not possible to select the technical areas in order to evaluate any routing problem in such areas that could appear. Also, depending in the zone or region where the network is developed, this “piece” of the Guifi.net network, works with different routing protocols, for example the zone of “Barcelona”, the main routing protocol which currently is being used is BGP protocol, in other areas such as “Girona”, the entire area uses OSPF routing protocol, and other areas such as “El Maresme” works with both protocols. The decision of which routing protocol is used in each zone depends on the network administrator of the area or on the skills of the people who will manage the network; also it should take into account the number of users that could use the network.

The following figure is an example of the currently configuration of the routing protocols of a concrete zone. The shown zone is the “Maresme” area, which belongs to the autonomous community of Catalonia; in that zone there are different routing protocols that are being used.

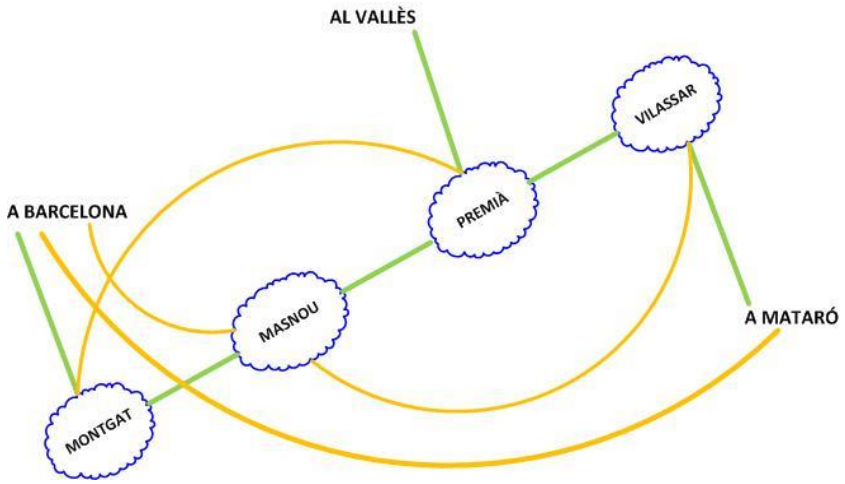



***Figure 2.1:*** *Example of link configuration in Guifi.net network*

As it can be seen, in this case there are OSPF zones for each city and each one is connected by a BGP link. In the figure presented, clouds represents OSPF zones of each city, green lines represents BGP links used to connect the clouds between them, finally the orange lines are the redundancies of OSPF zones. The redundancies of the zones are to make a network more reliable in front any problem that could occur in a network like this.

In Guifi.net network, in order to no create a networking problem the OSPF zones are “separated” from the rest of the zones of the network. In OSPF zones

is necessary to establish a nodes called Autonomous System Boundary Router, which are the responsible for routing external traffic to the other BGP AS, and route traffic internally the OSPF zone, this configuration will be explained in next section with more detail. For separate each routing zone, is applied an special configuration in each one of the Autonomous System Boundary Router (ASBR in advance) of the OSPF zone established in the piece of network, in order to configure an automatic firewall to not to provoke problems in the network.

### 2.1.1 BGP routing protocol

In BGP protocol is necessary to set Autonomous System (AS). Each AS includes his number and then propagates the route to the other AS's of the network, in order to update the routing table of the nodes and made a path of the network. This path made, it is an attribute of BGP called AS_Path.

AS_Path is the BGP attribute that keeps track of each AS when a route advertisement has passed through it. AS_Path is used by confederations and by exterior BGP (eBGP) to help prevent routing loops. When a BGP speaker propagates a route learned from another Update BGP speaker's message, this node modifies the AS_Path from the routes stored in the routing table in order to know exactly the location of the BGP speaker to which the route will be sent.

A brief summary of the update of the AS_Path attribute is the following:

- When a given BGP speaker advertises the route to an internal peer (iBGP), the advertising speaker not modifies the AS_Path attribute associated with the route.

- When a given BGP speaker advertises the route to an external peer (eBGP), the advertising speaker updates the AS_Path

The next figure shows the process of how the routes are propagated from one AS (in that case AS1) to the rest of the network, and how the AS_Path is changed in every hop which passes through a different AS:

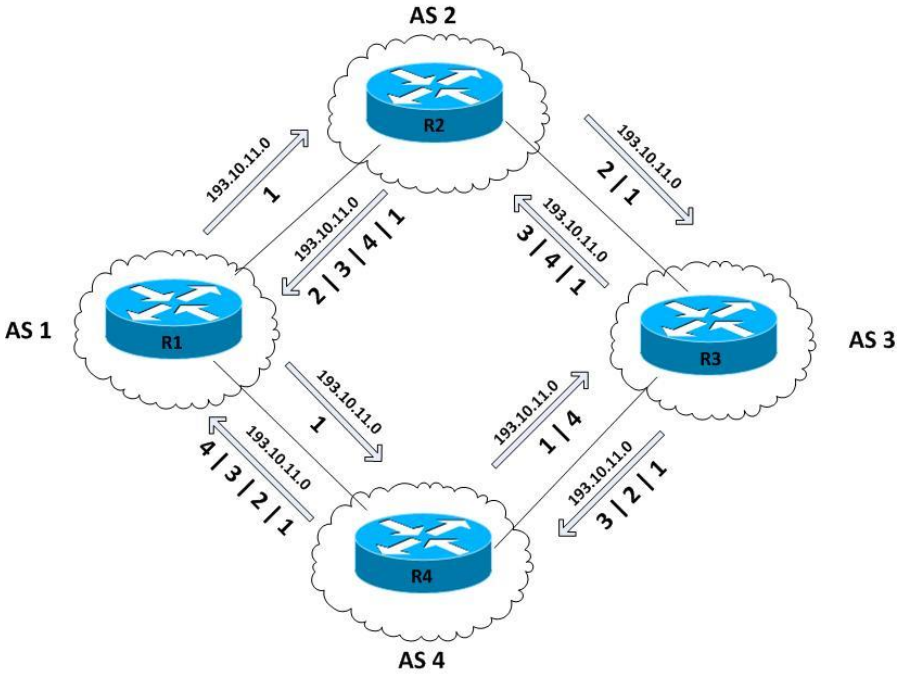



***Figure 2.2**: How the path of a network is transmitted to the different nodes*

As it can be seen in the above figure, AS1 is linked to AS2 and AS4, when AS1 speaks to the other AS's of the network, will originate a route that will track the following steps:

- The originating speaker (AS1) includes its own AS number in a path segment (AS_Path), AS_Path will be the attribute of Update messages sent to an external peer, these external peers are AS2 and AS4, as it can be seen in the figure.

- The originating speaker includes an empty AS_Path attribute in all Update messages sent to internal peers.

For example, in the example showed before, in case AS1 originates a route to AS3, that route will pass through different AS's, in that case, is it possible to pass through (AS1)→AS2→AS3 or also (AS1)→AS4→AS3. In both cases the routes passes across different AS's that only add their information about the Path, but for these AS's the content of the packets are not relevant, are like boxes, the only information that the AS's adds is the information about the path. Then the only thing that the node which is in AS1 knows, is if it sends a packet to the other AS, then the packet will arrive to the destiny, because routes have been made previously.

### 2.1.2 OSPF routing protocol

In OSPF protocol is not necessary to set Autonomous Systems like in BGP, in this protocol, a route are referred to networks, but is not possible to know if the path to follow will be so long or not. In OSPF protocol, packets are always sent to the next hop, and in these hops, the node will know how to route the packet to the correct destiny. This protocol is necessary to communicate the different nodes inside an AS, but this OSPF has some problems that will be explained with more detail in the next sections.

For routers that uses OSPF protocol and those routers belong to multiple areas, and connect these areas to the backbone area is it necessary to set elements called ASBR. Is a node which connects to different AS's and also interchange information with other nodes inside the same AS, then this node run more than one routing protocol at the same time. This node typically runs an exterior routing protocol (eBGP) to communicate with other nodes of different AS's, and an internal routing protocol (OSPF) to communicate with the node of the same AS. This nodes are used to distribute routes received from other, external AS's throughout its own autonomous system.

The next picture shown the border routers in a OSPF network where more than one routing protocol is running at the same time.

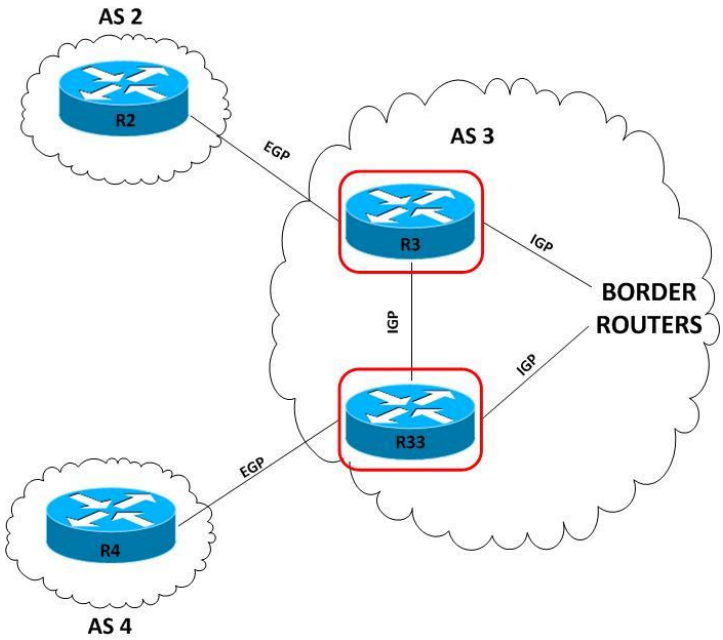



***Figure 2.3**: Border routers in a BGP and OSPF network*

## 2.2 Introduction to the problem

In a free, open an neutral network as Guifi.net, network problems can occur at any time, usually caused by users who want to join to the network, but this kind of users don't have enough knowledge in administration networks. And then by doing the action of adding or removing a node, this user unintentionally could cause a malfunction in the network that will affect other users in terms of connectivity and network performance.

Once this action is done, the network become to change and the problems in the network would appear. The first thing that happens in the network is the following: how OSPF protocol has higher preference in front BGP protocol, OSPF protocol changes the information (for example the attribute AS_Path) that contains the packets, and then these packets lose important information from the origin source. For that reason in the next chapters this mentioned problem will be analyzed in detail, then a solution to the problem will be given. The figure represents the BGP/OSPF connections that are currently in the network

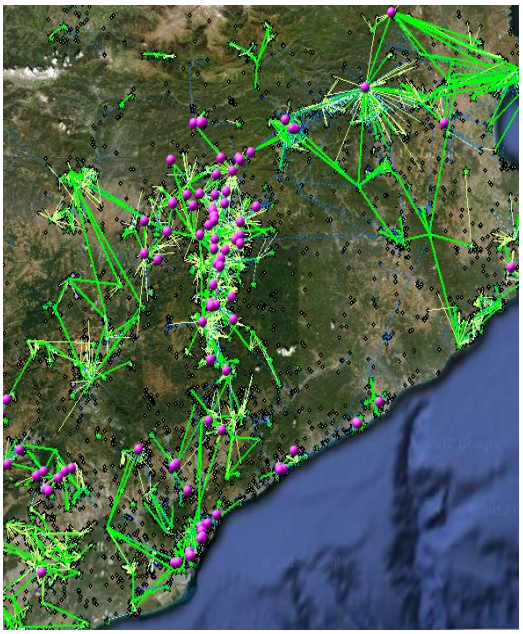


***Figure 2.4:*** *OSPF and BGP Guifi.net network*

### 2.2.1 Problems mixing OSPF and BGP protocols

The main problem that has Guifi.net network is the following; a packet is send from a node inside AS1 to another node that it is in AS4, but this packet has to pass thought an OSPF zone, as it could be the OSPF zone configured in AS3. As it was explained before, what happen in AS2 has to be invisible to AS1, but AS2 has to keep the AS_Path of AS1 and include the AS_Path of AS2, but when the packet go trough of an OSPF zone, this AS_Path is lost because OSPF protocol don't use and then don't store this kind of information.

Thus, the packet that was sent by AS1 is marked that was sent by AS3, provoking that when the packet arrives to the destiny (AS4) the node has wrong information, because it has the IP from AS1 and the AS information from AS3. Then the destiny node updates the information that has in the routing table with the new wrong information. Also, the problem becomes bigger because AS1 how receives wrong information, will send Update packets to the other nodes in order to update their routing tables.

As it can be seen in the figure above, once the update message are outside of AS3, always has the path of the AS3, changing the original information and becoming a problem in the network.

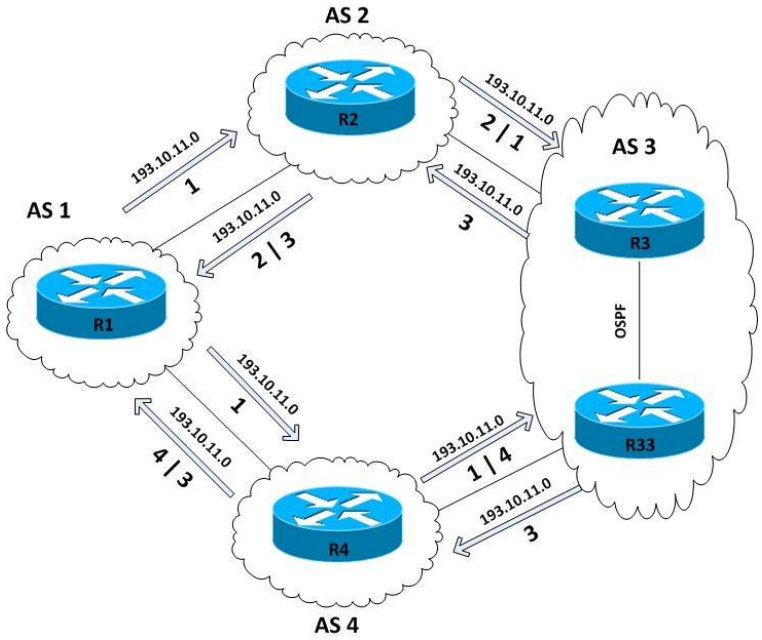



***Figure 2.5**: How the path of a network is transmitted to the different nodes*

Once the border route of AS3 has updated the information of the routing table, sent this new information to the rest of the nodes of the network, then the rest of the network receives the route from OSPF and change the information that have stored in the routing table for the new information received. This new information is propagated to the entire network, with the result that the all the nodes will have the same problem with the routing table in all the routers.

Then this problem about losing the AS_Path attribute in OSPF zones will be studied with more detail in the next chapters with the help of the emulations of the network.

# CHAPTER 3. NETWORK SOFTWARE EMULATOR

## 3.1 Introduction

Understanding computer networks without performing practical experiments is really difficult. Unfortunately, setting up a networking lab can be very expensive.

Netkit[7] is the result of the joint work of several people from the Computer Networks Laboratory of the Roma Tre University and from the Linux User Group LUG Roma. Netkit is an environment for setting up and performing networking experiments at low cost and with little effort. It allows emulating virtual network devices can be easily interconnected in order to form a network on a single PC. Networking equipment's are virtual but with many of the characteristics of the real ones, including the configuration interface. To emulate a network with Netkit is it necessary to write a simple file describing the link-level topology of the network to be emulated and some configuration files that are identical to those used in the networking tools.

## 3.2 Why Netkit

Using a network software emulator allows to the users the ability to create virtual environments that can be worked, studied and analyzed for experimentation, testing, etc. This kind of software forms an essential tool for experimental practice in research, development, training and integration of community networks.

Netkit is a software network emulation based on a Linux kernel virtualization. Each one of the nodes or virtual machines created, are accurate representations of the systems that can be found in production environments with the same features, configurations and communication protocols. Thus, the configurations of virtualized devices can be applied in actual devices. Also, is possible to emulate a network using the same software and configuration than in a real network. In addition, this tool is possible to be installed freely and, if it is necessary, make the necessary changes to solve any problems that may occur or even hold improvements deemed appropriate.

### 3.2.1 Netkit Features

Netkit is a lightweight network emulator based on open source software, with no impact on operational networks. Nekit provides to the users tools to support sharing preconfigured virtual labs with others. Netkit ships with a variety of supported networking tools and technologies. Should it be needed, additional software can also be installed inside the virtual machines in order to enable particular features that are required for specific experiments.

[7] http://wiki.netkit.org

# CHAPTER 4. HOW TO CREATE A “POLLASTRE DE RUTES”

## 4.1 Introduction

Once presented the scenario that cause the problems and the environment to develop the study, will be emulated a series of scenarios to reproduce these stated problems. These scenarios could be tested with the software introduced before. The scenarios will be focused on the technologies presented in previous chapters and its realization will emulate Guifi.net network.

The main objective of this chapter is by emulating a networking model with Netkit; test the network in order to observe how the routes loses the AS_Path attribute when an update message pass across an OSPF zone. It will also be observed how this change will affect to the rest of the entire network. Also during the test it will be observed how the network reacts to any possible change of topology that may occur, for example when a node or link crashes or when a new node or link is set up in the network.

## 4.2 Definition of “pollastre de rutes”

A “pollastre de rutes” is the colloquial name given in Guifi.net when a problem with routing protocols has been detected. After the problem have been announced, the network administrators of the affected area will try to determine the main reason that caused the problem and would isolate the problematic zone in order to not affect the performance of the entire network.

## 4.3 Network configurations

This section shows the commands that are needed to be added in the configuration files in order to have a good network performance. Furthermore in the Annex II, is it possible to see how an easy example of how to configure a network.

The developed tests are intended to be as realistic as possible of the current Guifi.net network. In order to reproduce the network, would be needed to configure several AS in order to emulate the Guifi.net network. In one of the AS configured would be needed two different Autonomous System Boundary Routers (ASBR in advance), to emulate the OSPF zone. Then inside the OSPF zone, the nodes that compound this area will be connected by OSPF links and the ASBR will connect to the other AS of the network by an eBGP link. The complete configuration will be described in the next section. The following figure represents the network configured in Netkit to execute the tests.

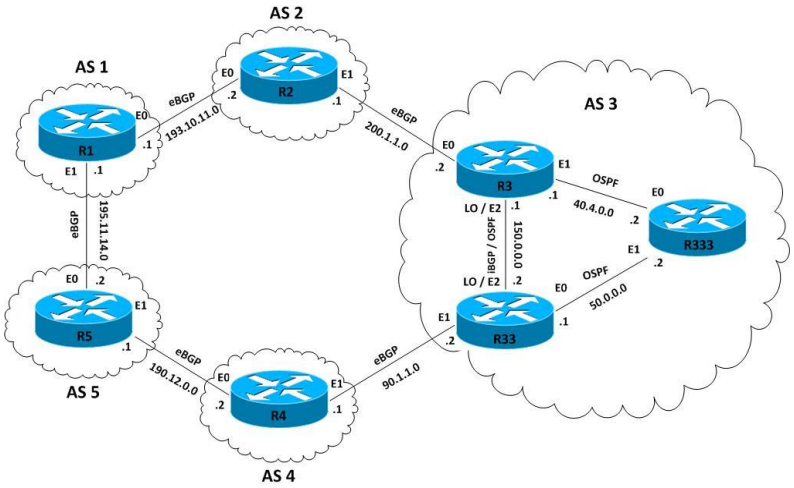



***Figure 2.1:*** *Initial configuration of the network, in AS3 could be OSPF or BGP the routing protocol between edge nodes*

For the previous network configuration, it was necessary to set certain commands in the configuration files. Without these commands configured in the files, the network performance was inadequate and could not be possible to perform the tests. The configured commands are the following:

### 4.3.1 Redistribution configuration

In order to redistribute the routing table that a router has stored in the kernel to rest of the nodes of the network is it possible with the command network and redistribute.

#### *4.3.1.1 Network command*

Network command is one possibility to advertise networks via BGP. Also is it possible to redistribute IGP into BGP. In this case IGP would be OSPF routing protocol.

***Table 4.1:*** *Network command example*

```
router bgp 3
network 200.1.1.0/24
network 40.4.0.0/24
```

#### *4.3.1.2 Redistribute command*

In the case of redistribution, this command dumps all internal routes into BGP. But this command has a drawback, is it needed to apply careful filtering to make sure that are send to the routes that are needed to be advertised.

***Table 4.2:*** *Redistribute command example*

```
router bgp 3
neighbor 90.1.1.1 remote-as 4
neighbor 90.1.1.1 description (Virtual) Router 4 of AS4
redistribute connected
```

### 4.3.2 No synchronization configuration

One problem that will be notice when this network is set up is not possible to see the BGP entries in the routing table. This is a synchronization issue because BGP does not put these entries in the routing table and does not send the entries in BGP updates due of a lack of synchronization with IGP.

Applying the problem to the network causes that R3 does not synchronize with OSPF because of the difference in masks, but this action also is necessary on R33 for the same reason, also disabling synchronization allows BGP to converge more quickly, but it might result in dropped transit packets

***Table 2.3:*** *No synchronization command example*

```
router bgp 3
no synchronization
network 200.1.1.0/24
```

To avoid the problem explained is it necessary to include the command “no synchronization” in the configuration files of R3 and R33.

### 4.3.3 Passive interface configuration

Also another problem will happen in the configuration mentioned before, is it necessary to enable OSPF on Eth0 of R3 to make it passive, in order to reach the outside network.

To make this possible is it necessary to configure in the OSPF configuration file the following command “passive-interface EthX” being X the interface that will reach the outside network.

***Table 4.4:*** *Passive-interface command example*

```
router ospf
passive-interface eth0
network 40.4.0.0/24 area 0.0.0.0
redistribute connected
```

This command allows the node to know about the next hop 200.1.1.2 via IGP. If this command is it not set, routing loops may occur.

### 4.3.4 Update-source configuration

In this case in order to configure iBGP between the two border routers, it has been used the following configuration:

***Table 4.5:*** *Update-source command example*

```
router bgp 3
neighbor 2.2.2.2 remote-as 3
neighbor 2.2.2.2 update-source 1.1.1.1
neighbor 2.2.2.2 description r33
```

```
router bgp 3
neighbor 1.1.1.1 remote-as 3
neighbor 1.1.1.1 update-source 2.2.2.2
neighbor 1.1.1.1 description r3
```

The previous configuration belongs to the border routers R3 and R33. In order to configure iBGP session between the loopback interfaces of these two routers is it necessary to use the command “update-source”, because this command enables any operational interface, including the loopback interface, can be specified to be used for establishing TCP connections. Without this command, the TCP session will use the IP address of the closest interface and the neighbor will reject the incoming TCP packet as it’s not coming from a recognized BGP neighbor. This command also allows the routers running BGP with multiple links between them to load balance over the available paths.

The complete configurations files for the network are included in the Annex III.

## 4.4 Emulating “Pollastres de rutes”

This section will explain the tests done to reproduce and test the problems that the network has when different routing protocols are working in the same network. The purpose of these is to compare how works the different routing protocols, BGP and OSPF, between different and inside the same AS, and how the routing table changes using one or other routing protocol without stopping the functionality of the network.

### 4.4.1 First scenario: BGP network

In AS3, have been configured the ASBR, R3 and R33. These border routers works inside the AS with BGP and also OSPF. BGP protocol has been configured in the loopback interfaces, with this configuration these routers are able to make an iBGP connection between them and exchange their routing tables. OSPF protocol has been configured in other interfaces to run the AS3 entirely with OSPF.

To finalize the configuration, also this two border routers are connected to external AS's, AS2 and AS4. These external AS's are connected to other different AS's, AS1 and AS5 in order to complete the ring configuration that was presented before and cause the problem of "pollastre de rutes".

Once configured the lab, the network was tested and the results were stored in order to compare them with the results obtained with OSPF test protocol. The network topology used for the first tests is the following:

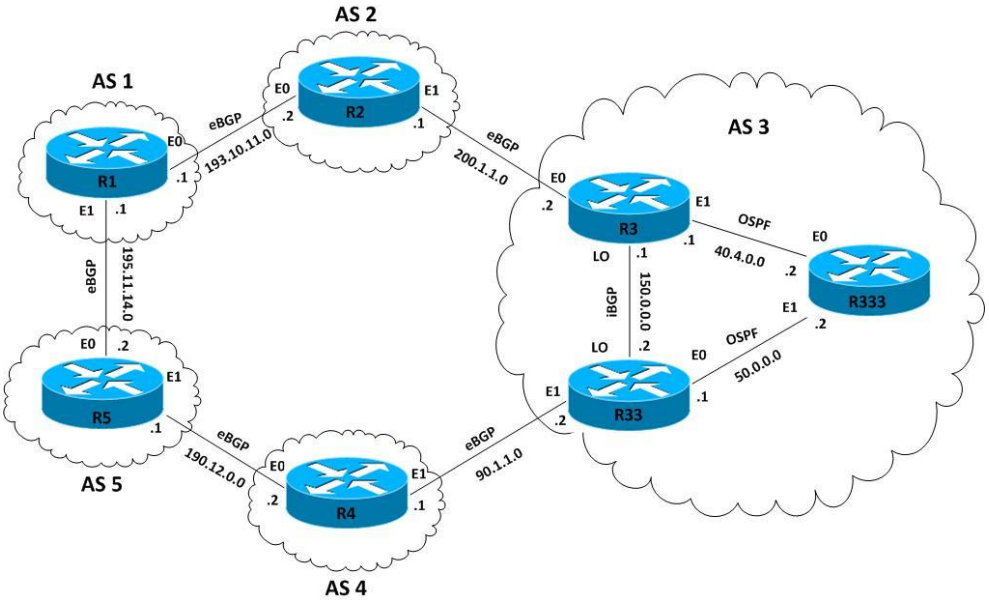



***Figure 4.2:*** *Network configuration with BGP as routing protocol in the edge nodes of AS3*

As it can be seen in the above figure, in AS3 the loopback interface between border routers R3 and R33 makes an iBGP link, then running as IGP between R3 and R333 and R33 and R333 is OSPF protocol, to provoke that when the routes has to pass thought AS3 will lose the AS_Path attribute. As it was indicated before for the rest of the networks the links done between the different AS's are done by eBGP protocol.

The complete configuration files of the scenarios are placed in Annex III

### 4.4.2 Results of the BGP networking

As the result of using this topology, is it possible to view the routing table and how a packet is it transmitted around the AS's of the network.

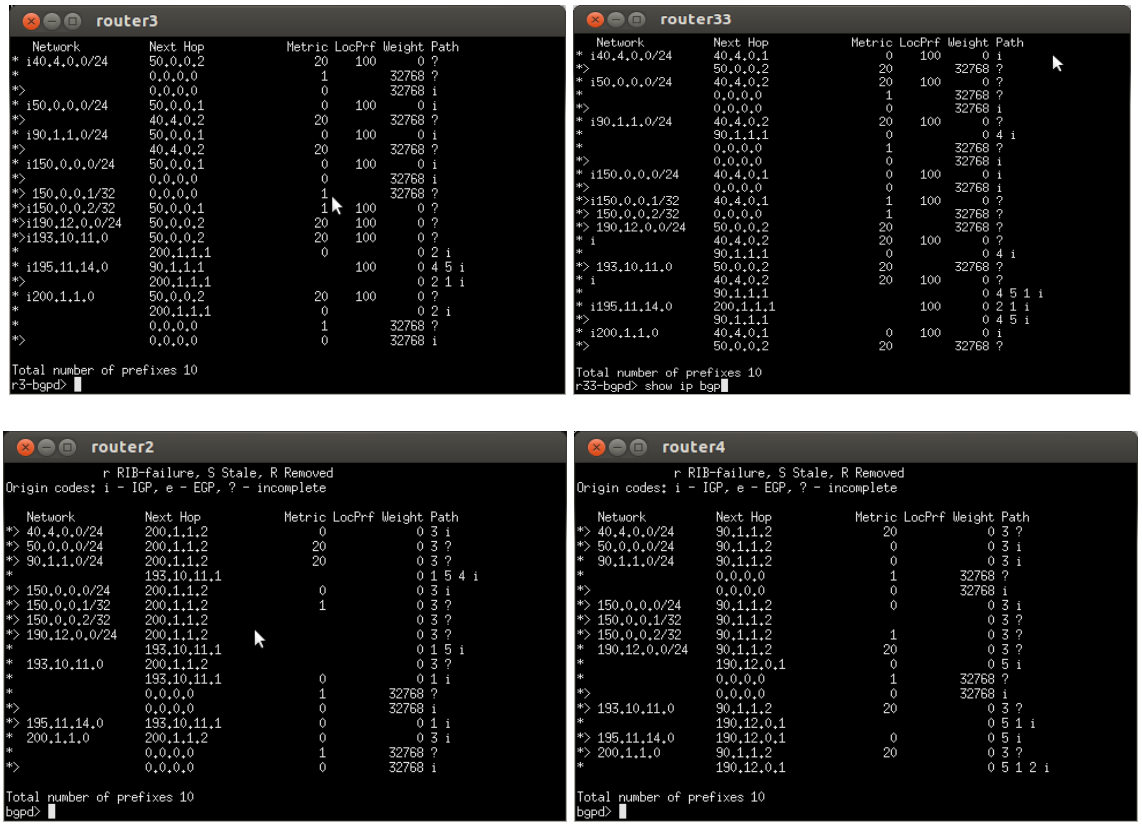



***Figure 4.3:*** *Routing tables from the edge nodes R3 and R33, and routing tables from R4 and R2 while BGP protocol is running between edge nodes of AS3*

As it can be observed in the routing tables, between R3 and R33 nodes, is established an iBGP link, which in in the routing table is indicated by an "i" before the number of the network. Due of this iBGP link, the two border routers are able to interchanges their routing tables of each one and propagate any change made on the router to the entire network.

Also is it possible to spot the following, when a network passes through AS3, the path of the route is marked with a "?". This symbol means that the route is incomplete because the router has lost the AS_Path of the network. As it was commented before this is due the OSPF link in AS3. In that case the AS2 left active the route that comes incorrectly originated by the AS3, and then notifies this route to AS1, changing AS1 their routing table and being part of the problem. Then AS1 will also receive this wrong information from AS4 and AS5 making the "pollastre de rutes" inside the ring network.

Despite this incomplete path, the network doesn't loss connectivity, but has lost an important attribute as the AS origin, then the routing table has wrong IP origin information, notifying that the same network is generated in two different

AS, then the node when has to choose the path, will choose the route with shorter AS_Path.

### 4.4.3 Second scenario: OSPF network

Once the iBGP test are done is it possible to show what happen when suddenly the network has a great change, how can be a change of the routing protocol inside a determined AS, in this case changing from iBGP protocol inside the AS3 to a OSPF protocol.

To run the OSPF emulation is it necessary to break the iBGP link, this can be done by getting down the loopback interfaces, and then is it necessary to break the running BGP protocol between R3 and R33 inside AS3 and change the routing protocol for the OSPF protocol.

In the OSPF case is not necessary to set any additional parameter in the border routers, like in BGP case. In that case, in order to redistribute default routes in to BGP, it has to be used the network statement.

***Table 4.6:*** *OSPF configuration for R3 node*

```
router ospf
passive-interface eth0
network 150.0.0.0/24 area 0.0.0.0
network 40.4.0.0/24 area 0.0.0.0
redistribute connected
```

The network topology used to do this test is the topology of the figure as follows:

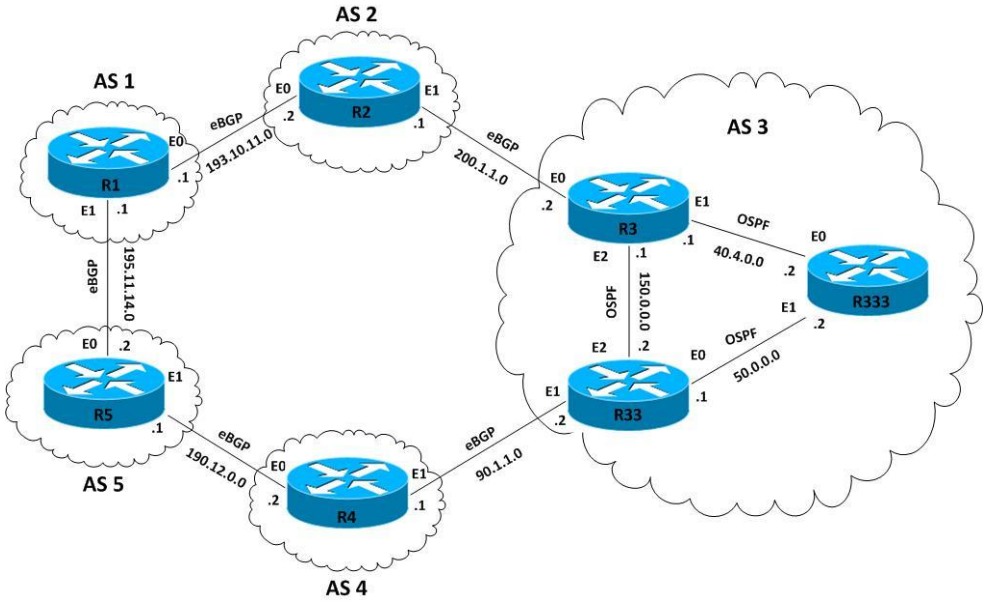



***Figure 4.4:*** *Network configuration with OSPF as routing protocol in the edge nodes in AS3*

As it can be seen in the figure, in AS3 between the border routers R3 and R33 is running OSPF, running OSPF protocol in the entire AS3 to cause the loss of the AS_Path. For the other nodes of the network, the links done between the external AS's are done by BGP protocol.

Once the OSPF protocol is working instead BGP protocol is it possible to view the changes done in the network referenced to the routing table.

### 4.4.4 Results of OSPF networking

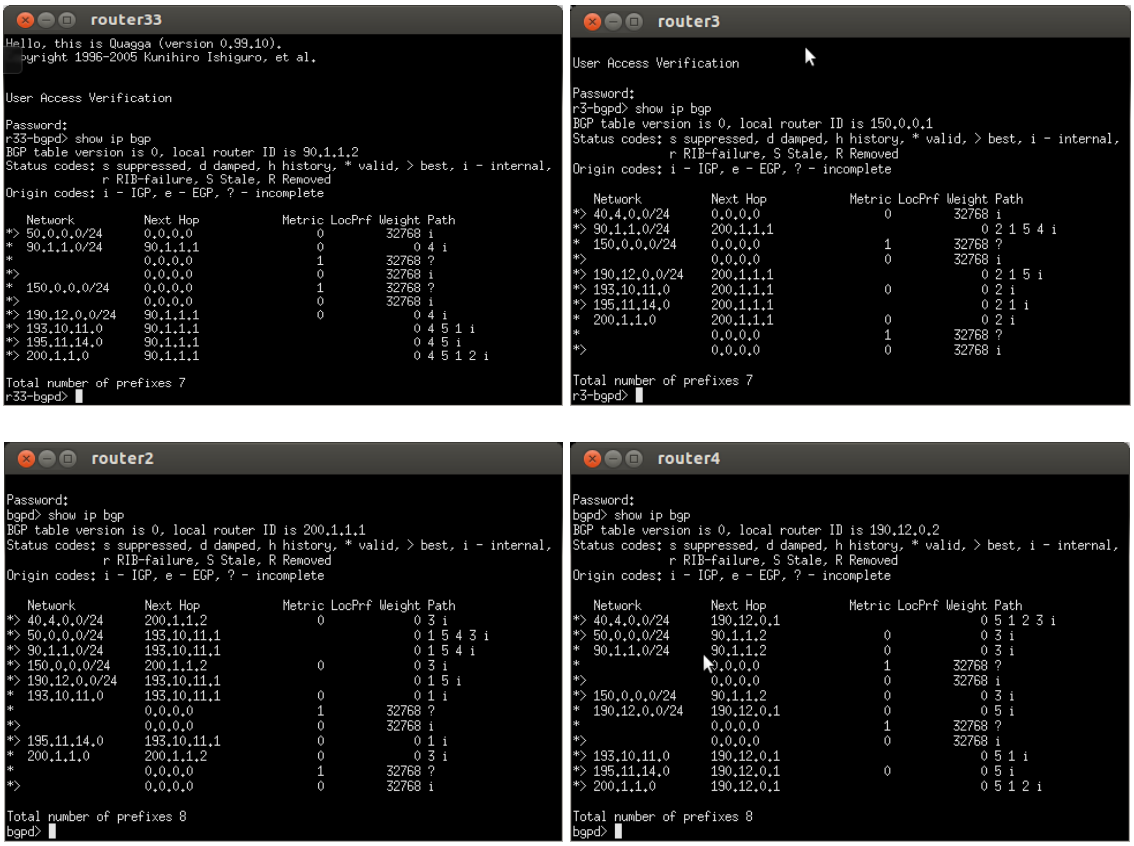



***Figure 4.5:*** *Routing tables from the edge nodes R3 and R33, and routing tables from R4 and R2 while OSPF routing protocol is running in edge nodes of AS3*

The first thing that can be observed in the routing tables is the IGP link that the border routes has with the iBGP link, have been disappeared, also the size of the routing tables have been decreased because the ASBR don't exchange information between them.

Due of this iBGP connection between the two border routers is no longer available; R3 and R33 do not exchange the routing tables between them and the network loses a "path" to reach the network. Because in an area where is performed by OSPF protocol, routers use the least cost path to the destination.

In OSPF, the routing between areas is the following:

1. Routers forward the packet along the least cost path to the nearest border router.
2. Border routers forward the packet along the least cost path to the nearest border router connected to the area that contains the IP destination address.

With the test completed, is it possible to achieve the following conclusion; when a connection is established between two borders routers in the same AS, this connection has to be made with BGP protocol. Without this connection, the nodes are not able to interchange their routing tables and then are not possible to have connectivity to the entire network.

## 4.5 Broken links and crashed nodes

In a real network is probably that a node or link could crash, because may have a cut current in the node, malfunction of one of the nodes, bad link-level, etc. In this section will be tested the network if a BGP node crashes outside AS3 while OSPF protocol is running and how the network reacts to these kinds of actions.

### 4.5.1 Third scenario: Breaking links in a network

To prepare the scenario in order to do the test, is it has been decided to break links in the node R5 in AS5, after doing that, the packets interchanged between the nodes will be captured with the software Wireshark which is a packet analyzer. With this information is it possible to show which kind of packets are send when one node crash in a network.

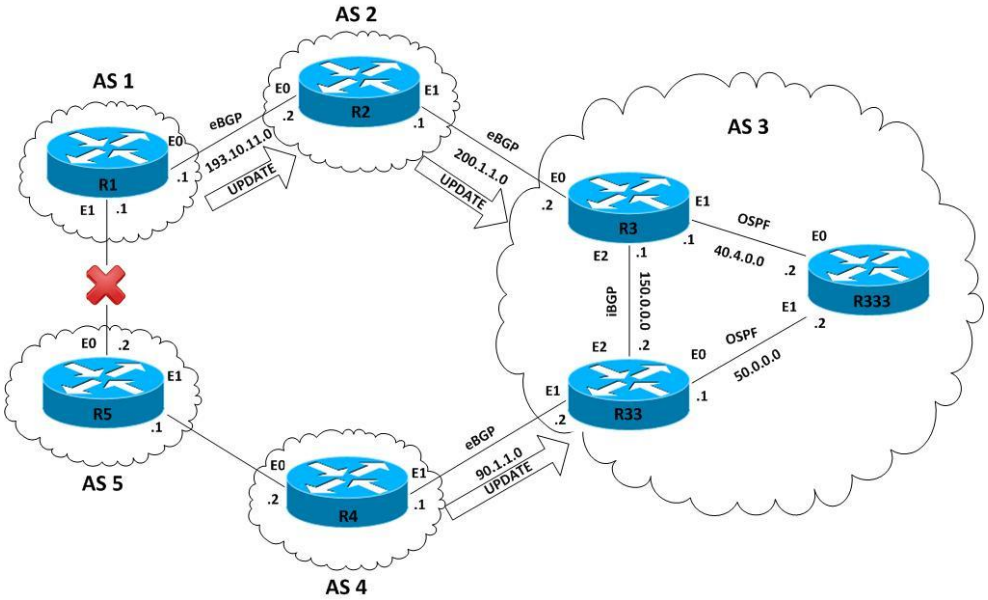



***Figure 4.6:*** *How the network converge after a break link*

As it can be view in the above figure, once one of the links of the node R5 is broken, the other routers exchange their routing information in order to update the routing table and to not send packets to this route where the node R5 was before.

In R2 Eth0, R3 Eth1 and R33 Eth1 will be captured the packets by the Wireshark software. As it was said before it will possible to capture the update packets sent to the other nodes in order to change their routing table due to the change in the network.

### 4.5.2 Fourth scenario: Crashing nodes in a network

Also is it possible to see what happen in the network when a node crashes. For running that test, it has been decided to crash the R5 node. Then the network has to converge in order to update the information that has in the routing tables.

For crashing the R5 node, Netkit allows to halt a router by the command vhalt, once the node is halted, as it was indicated before, the network has to update the information and then Updates packets will be sent between the nodes. Then the network behavior is it similar when a node crash as when a link crash.

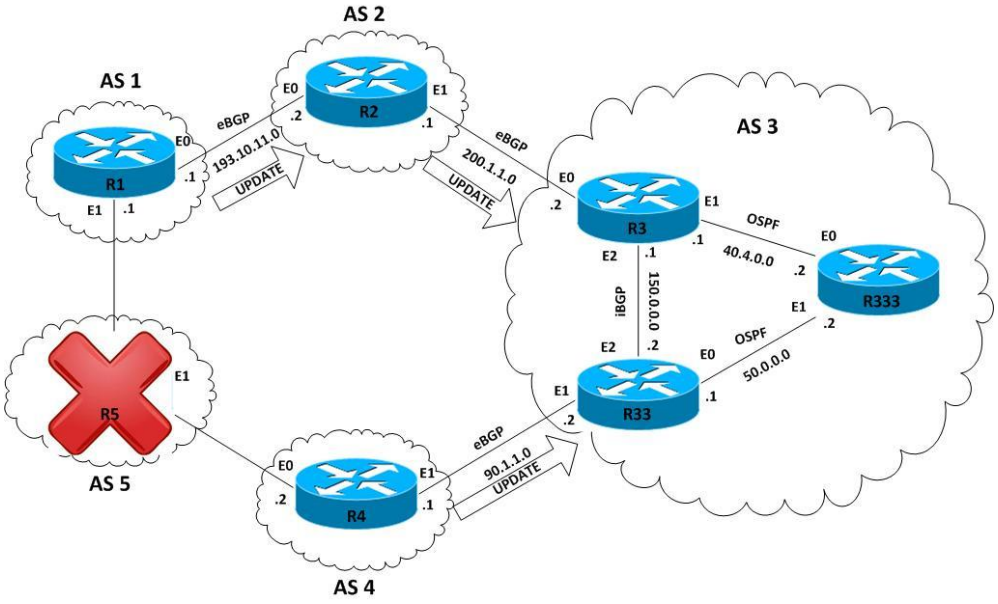



***Figure 4.7:*** *How the network converge after a crashed node*

### 4.5.3 Results of crash or nodes links in a network

The following routing table shows how the routers change their routing table in order to establish new routes due to the R5 node crash

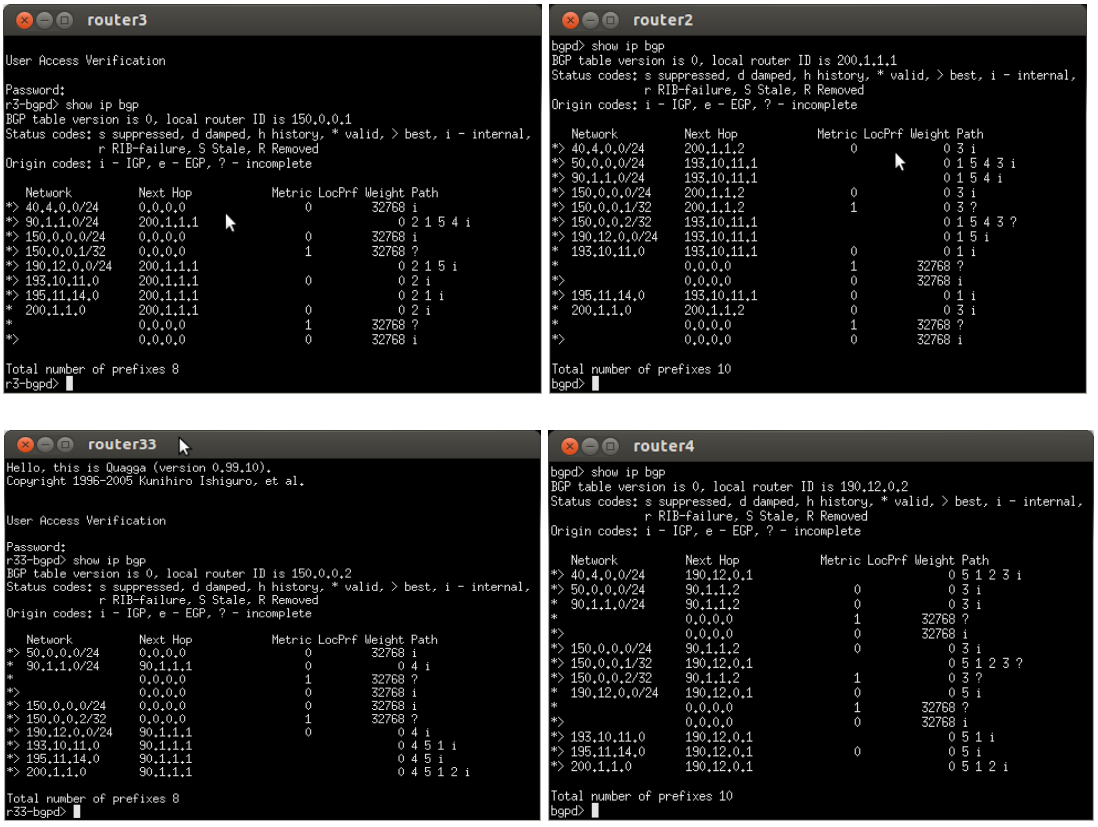



***Figure 4.8:*** *Routing tables from the edge nodes R3 and R33, and routing tables from R4 and R2 after the node R5 have been crashed.*

Once the interfaces of the node R5 have been down and the traffic that was generated have been captured, is possible to detect how the nodes send the update information packet to the rest nodes of the network. Also is possible to view, how the information has changed, the routes that belongs to R5 are not possible to be reached, and also how the ring configuration has been broken, the IGP connection between R3 and R33 also have been disappeared.

As it can be observed in the following figure, it represents and Update packet capture with Wireshark[8], this packet includes information from the network, announcing to the other nodes of the network that the networks 40.4.0.0 and 150.0.0.1 are not available due to the crash of the node.

[8] http://www.wireshark.org/

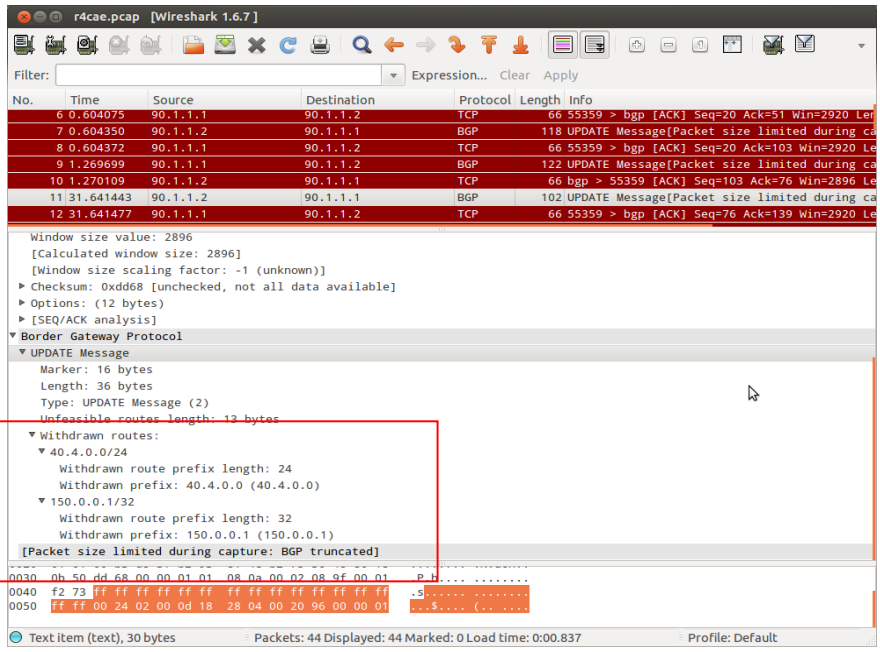



***Figure 4.9:*** *Wireshark capture showing the Update Packets sent from the other nodes in order to update the routing table*

As conclusion for these tests can be obtained two different cases, when in the network the nodes crashes (Node from On to Off) and when in the network a link between nodes breaks (Link from On to Off).

- Node On-Off: Neighbor nodes would erase all these routes from this node in their tables, which has as next hop the crashed node, then the nodes has to recalculate new routes. The neighbor nodes of the crashed node which are inside the same AS, will know the event instantly, while those which linked by an eBGP link, must wait until the next keepalive message. The new route configuration is propagated in the network until in all routing tables are no routes that has to through the damaged AS or through the router that causes the anomaly.

- Link On-Off: It is similar case to the previous one, but instead of crashing a node crashes an eBGP link. In that case nodes connected between them by this link are instantly erased in the routing table and then the nodes have to recalculate routes. The nodes transmit the new routes calculated and the process ends when no route is in the routing table which AS_Path link has been disabled.

## 4.6 Restoring links and nodes after a crash

Following the previous test done, will be analyzed the network when a node or link which was crashed is restored again in the network.

### 4.6.1 Fifth scenario: Restoring broken links in a network

In this case to restore a link can be made easily “upping” the interfaces of the router. This will be made by typing in the console of the router the following command: “ifconfig EthX” up with this command the interfaces will be restored and then the nodes will recover the visibility to these interfaces.

Once the interfaces are restored, will be analyzed what happen in the routing table, where previously the routes that pass along R5 where erased from the routing table. At this point will be analyzed if the routes will be restored how at the beginning of the emulation.

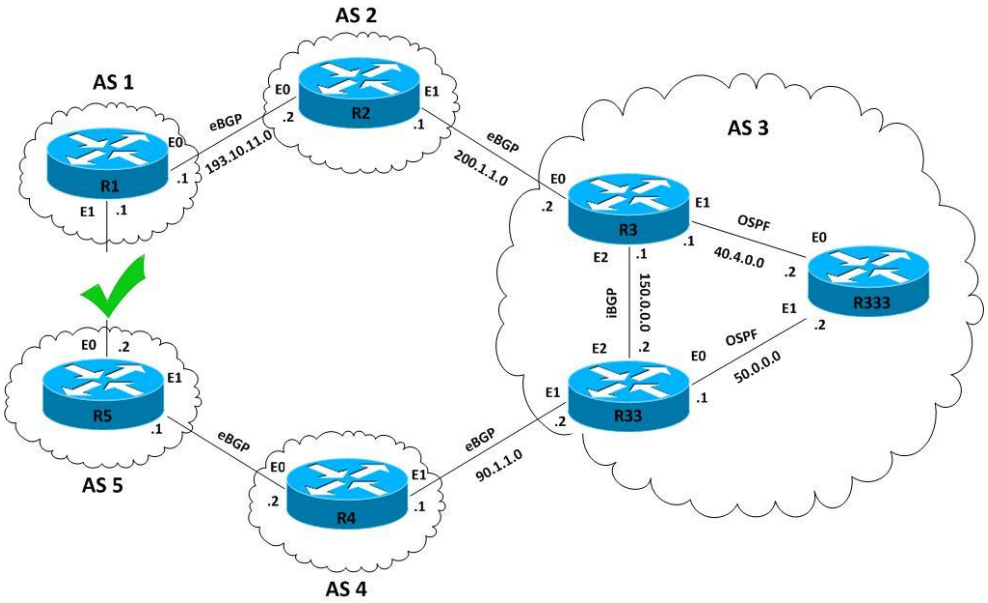



***Figure 4.10:*** *How the network converge after a link is recovered from a crash*

In R2 Eth0, R3 Eth1 and R33 Eth1 will be captured the Wireshark packets to observe if the update packets is sent to the other nodes of the network, so these nodes could change the information that has stored in the routing table.

### 4.6.2 Six scenario: Restoring crashed nodes in a network

In this test the node R5 will be recovered from a crash by typing the command “vstart R5” in Netkit. Once doing this will be possible to capture information sent by the nodes of the network, in order to update their information in the routing tables.

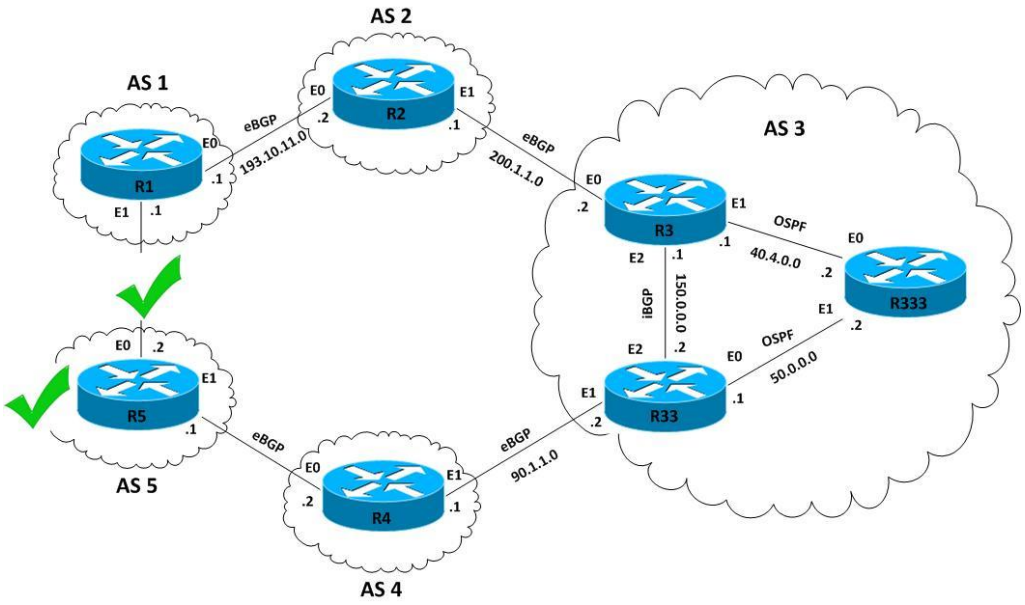
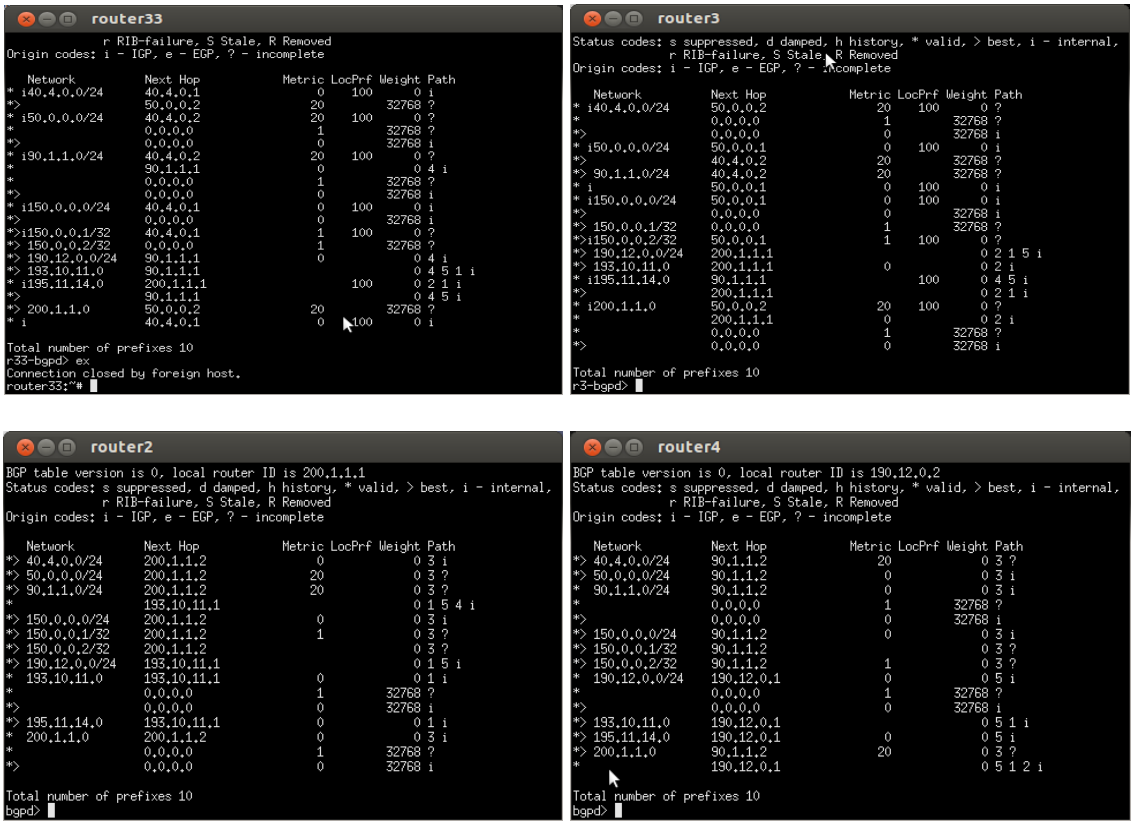



***Figure 4.11:*** *How the network converge after a node is recovered from a crash*

### 4.6.3 Results of nodes and links restored in a network



router33

```
            r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i40.4.0.0/24     40.4.0.1                 0    100      0 i
*>                 50.0.0.2                20         32768 ?
* i50.0.0.0/24     40.4.0.2                20    100      0 ?
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i
* i90.1.1.0/24     40.4.0.2                20    100      0 ?
*                  90.1.1.1                 0           0 4 i
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i
* i150.0.0.0/24    40.4.0.1                 0    100      0 i
*>                 0.0.0.0                  0         32768 i
*>i150.0.0.1/32    40.4.0.1                 1    100      0 ?
*> 150.0.0.2/32    0.0.0.0                  1         32768 ?
*> 190.12.0.0/24   90.1.1.1                 0           0 4 i
*> 193.10.11.0     90.1.1.1                             0 4 5 1 i
* i195.11.14.0     200.1.1.1                     100     0 2 1 i
*>                 90.1.1.1                             0 4 5 i
*> 200.1.1.0       50.0.0.2                20         32768 ?
* i                 40.4.0.1                 0    100      0 i

Total number of prefixes 10
r33-bgpd> ex
Connection closed by foreign host.
router33:~#
```

router3

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
* i40.4.0.0/24     50.0.0.2                20    100      0 ?
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i
* i50.0.0.0/24     50.0.0.1                 0    100      0 i
*>                 40.4.0.2                20         32768 ?
*> 90.1.1.0/24     40.4.0.2                20         32768 ?
* i                 50.0.0.1                 0    100      0 i
* i150.0.0.0/24    50.0.0.1                 0    100      0 i
*>                 0.0.0.0                  0         32768 i
*> 150.0.0.1/32    0.0.0.0                  1         32768 ?
*>i150.0.0.2/32    50.0.0.1                 1    100      0 ?
*> 190.12.0.0/24   200.1.1.1                             0 2 1 5 i
*> 193.10.11.0     200.1.1.1                0           0 2 i
* i195.11.14.0     90.1.1.1                      100     0 4 5 i
*>                 200.1.1.1                            0 2 1 i
* i200.1.1.0       50.0.0.2                20    100      0 ?
*                  200.1.1.1                0           0 2 i
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i

Total number of prefixes 10
r3-bgpd>
```

router2

```
BGP table version is 0, local router ID is 200.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 40.4.0.0/24     200.1.1.2                0             0 3 i
*> 50.0.0.0/24     200.1.1.2               20             0 3 ?
*> 90.1.1.0/24     200.1.1.2               20             0 3 ?
*                  193.10.11.1                            0 1 5 4 i
*> 150.0.0.0/24    200.1.1.2                0             0 3 i
*> 150.0.0.1/32    200.1.1.2                1             0 3 ?
*> 150.0.0.2/32    200.1.1.2                              0 3 ?
*> 190.12.0.0/24   193.10.11.1                            0 1 5 i
*  193.10.11.0     193.10.11.1              0             0 1 i
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i
*> 195.11.14.0     193.10.11.1              0             0 1 i
*  200.1.1.0       200.1.1.2                0             0 3 i
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i

Total number of prefixes 10
bgpd>
```

router4

```
BGP table version is 0, local router ID is 190.12.0.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 40.4.0.0/24     90.1.1.2                20             0 3 ?
*> 50.0.0.0/24     90.1.1.2                 0             0 3 i
*  90.1.1.0/24     90.1.1.2                 0             0 3 i
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i
*> 150.0.0.0/24    90.1.1.2                 0             0 3 i
*> 150.0.0.1/32    90.1.1.2                               0 3 ?
*> 150.0.0.2/32    90.1.1.2                 1             0 3 ?
*  190.12.0.0/24   190.12.0.1               0             0 5 i
*                  0.0.0.0                  1         32768 ?
*>                 0.0.0.0                  0         32768 i
*> 193.10.11.0     190.12.0.1                             0 5 1 i
*> 195.11.14.0     190.12.0.1               0             0 5 i
*> 200.1.1.0       90.1.1.2                20             0 3 ?
*                  190.12.0.1                             0 5 1 2 i

Total number of prefixes 10
bgpd>
```

***Figure 4.12:*** *Routing tables from the edge nodes R3 and R33, and routing tables from R4 and R2 after the node R5 have been recovered from the crash.*

How network configuration has been restored, the first thing that could be observed in the routing tables is the iBGP connection between R3 and R33 also have been restored.

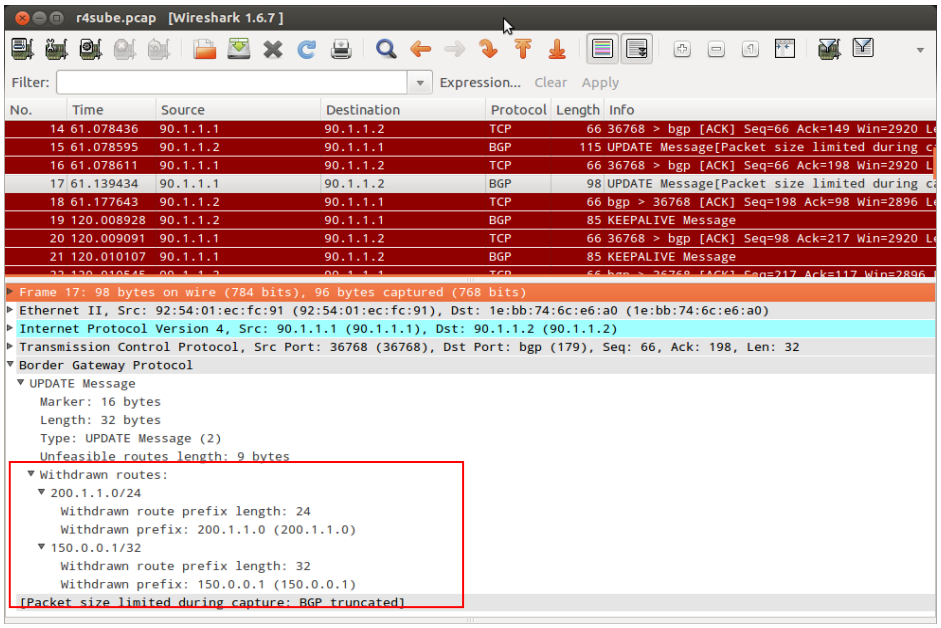



***Figure 4.13:*** *Wireshark capture showing how the update packets are sent to the rest of the nodes in order to update the routing table*

As it can be observed in the figure of the software Wireshark, this Update packet announces to the rest of the network, that the networks 200.1.1.0 and 150.0.0.1 are available. Also, the conclusions that can be obtained for these tests are the following:

- Node Off-On: In a stable network at a given instant, a new node joins to the network and the neighbors to which it connects are aware of their presence. These nodes will send to this new node their best routes to reach known destinations. Then, the new node configures its routing table and advertises their routes to the network neighbors. In the network best paths provided by this new node are transmitted to the rest of the network.

- Link Off-On: a new eBGP link is established to connect two AS's. The affected nodes will interchange their routing tables. This will cause a recalculation in their respective tables, generating new announcements in the network.

# CHAPTER 5. POSSIBLE SOLUTIONS TO AVOID "POLLASTRE DE RUTES"

## 5.1 Introduction

In this chapter will be analyzed the different methods to try to avoid the problem of "pollastre de rutes" mentioned in before chapters.

## 5.2 Location of the solutions

Once the problem is presented and evaluated how critical is, is necessary choosing in which nodes to be has executed the different solutions.

In the presented case, where is wanted to evaluate the problem in order to detect the "pollastres de rutes" the solution has to be defined in a node where is possible to control the entire network. Whereas has been decided to execute the solution in the ASBR's of AS3, which are the ones which has a total vision of the network and has the OSPF link which creates the routing protocol in the network. Also the OSPF zones are the responsible of the problems of routing protocols to the rest of the network; hence all the solutions will be focused in this piece of the network.

## 5.3 Script solution

The main purpose of this script is to identify potential "pollastres de rutes" on the network. This script will access to the routing tables of the nodes using two different methods and then compare the information obtained to determine if there is a problem in the network. One of these methods is the SNMP protocol, which allows remote access to the routing tables of neighbor nodes. The other method is by "vtysh" command, which allows downloading the content of the routing tables stored in the kernel of the nodes. Once downloaded both information, will be analyzed the routing tables obtained from each node and determine if there are different AS information between these two files. If is identified any network difference, it will be automatically notified to the network.

The problem is that BGP only sends information when a change is done in the network, then sends an "Update" packet in order to update the routing table and send this information to the neighbors to allow them to update their routing tables. These "update" packets could be send by a change of topology, broken link or any other change that has to be done in the routing tables. However OSPF does not work as BGP, in OSPF networks information is send periodically, by default this information is send each 10 seconds. In these announcements any change in the network is included. Therefore the script has to be executed when a change is done in the network, i.e. when an "update" packet is send, because any change done in the network automatically will be announced by BGP and OSPF protocols. Doing this will avoid flooding the network with unnecessary requests and nodes will not suffer overload of request about their routing tables.

### 5.3.1 How works the script?

#### 5.3.1.1 *Obtaining routing tables from the kernel*

Netkit software allows obtain the routing tables using the command “vtysh”, with this command is it possible to obtain a node's routes from the kernel. In order to obtain the routing table from the BGP configuration it has to be used the following command:

***Table 5.1:*** *Obtaining Routing tables stored in the kernel of the node by vtysh command*

```
vtysh -e "show ip bgp" > /hostlab/file.txt
```

#### 5.3.1.2 *Obtaining routing table from a neighbor node*

Also Netkit as being a network emulator allows the “SNMP” command execution, being possible using this command to obtain the routing tables of neighbor nodes. The only requisite to make is to giving the necessary permissions to the different groups that compound this protocol.

In order to give and configure the different permissions is it necessary to modify the file snmp.conf. Once modified the snmp.conf file is it possible to obtain the routing table from a neighbor node by using the following command:

***Table 5.2:*** *Obtaining Routing tables stored of the node by SNMP command*

```
snmptable -Cb -Cw 80 90.1.1.2 public ipRouteTable
```

The next figure shows the data that the SNMP command extract from the node selected.

***Table 3:*** *SNMP table from the node*

```
       Dest IfIndex Metric1 Metric2 Metric3 Metric4     NextHop     Type
   40.4.0.0       0       0      -1      -1      -1     50.0.0.2   direct
   50.0.0.0       2       0      -1      -1      -1     0.0.0.0    direct
   90.1.1.0       3       0      -1      -1      -1     90.1.1.1   direct
 190.12.0.0       4       0      -1      -1      -1     90.1.1.1   indirect
193.10.11.0       1       0      -1      -1      -1     50.0.0.2   indirect
195.11.14.0       0       0      -1      -1      -1     90.1.1.1   indirect
  200.1.1.1       0       0      -1      -1      -1     50.0.0.2   indirect

SNMP table ip.ipRouteTable, part 2

 Proto Age          Mask Metric5           Info
 local   0 255.255.255.0      -1 .ccitt.nullOID
 local   0 255.255.255.0      -1 .ccitt.nullOID
 local   0 255.255.255.0      -1 .ccitt.nullOID
 local   0 255.255.255.0      -1 .ccitt.nullOID
 local   0 255.255.255.0      -1 .ccitt.nullOID
 local   0 255.255.255.0      -1 .ccitt.nullOID
 local   0 255.255.255.0      -1 .ccitt.nullOID
```

Once the files are extracted from the nodes and with the help of the script done is it possible to compare the routing tables of the own node and the neighbor node and determine if the routes stored in the kernel of the node and the routes obtained from the SNMP table had different origins, in case the script determines that there are different beginning and different routes will show message altering that there is a problem with the routing in the network and the network should be reviewed to no originate more problems.

## 5.4 Applying filters

### 5.4.1 Route filtering

Filtering is a possibility to restrict the routing information that a particular router learns or advertises. The filter consists of an access list that is applied to updates to/from a neighbor, so packets received in such nodes must follow a set of policies. By this method, is also possible to avoid some of the "pollastres of rutes".

In that case of the figure above, R333 is originating network 50.0.0.0 and sending it to R3 and R33. To prevent R333 propagating routes updates from the network 50.0.0.0 to the other AS's of the network, is it possible to apply an access list to filter those updates when R333 exchanges updates.

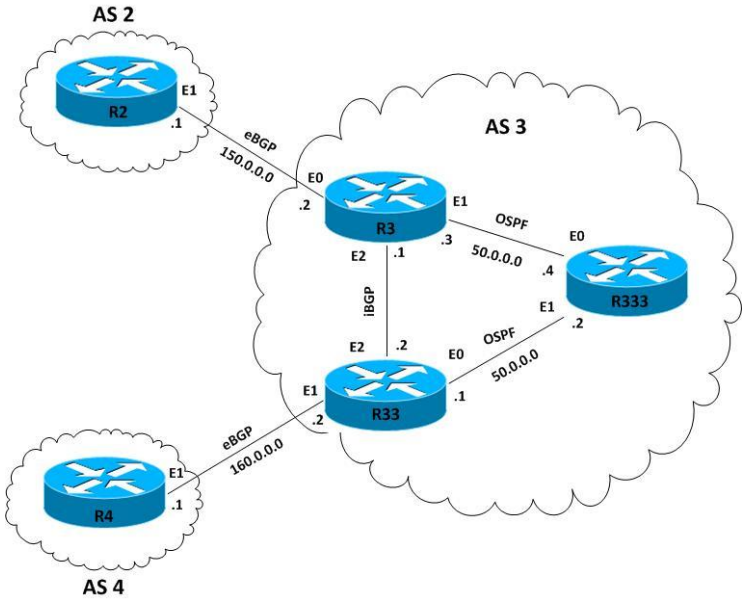



***Figure 5.1:*** *Prefix Filtering test*

To filter this network 50.0.0.0, it has to be configured in the border routers R3 and R33 the following configuration:

***Table 5.4:*** *Prefix filtering configuration for R3 node*

```
router bgp 3
neighbor 160.0.0.1 remote-as 4
neighbor 160.0.0.1 distribute-list 1 out
neighbor 160.0.0.1 distribute-list 1 in
!
access-list 1 deny 50.0.0.0 255.255.255.0
access_list 1 permit 0.0.0.0 255.255.255.255
access_list 1 deny 0.0.0.0 255.255.255.255
```

With the combination of the neighbor distribute-list router configuration command and access list 1 prevents R333 from propagating routes for network 50.00.0.0 when it sends routing updates to neighbor R3 or R33.

### 5.4.2 AS_Path filtering

Routes can be filtered based on AS-Path values, using a command called as-path access-list.

***Table 5.5:*** *AS_Path filtering configuration in R3 and R33 nodes*

```
neighbor 200.1.1.1 remote-as 2
neighbor 200.1.1.1 route-map setlocalpref in
neighbor 200.1.1.1 description (Virtual) Router 2 of AS2
!
route-map setlocalpref permit 10
set local-preference 200
```

```
neighbor 90.1.1.1 remote-as 4
neighbor 90.1.1.1 route-map localonly in
neighbor 90.1.1.1 description (Virtual) Router 4 of AS4
!
ip default-network 90.1.1.0
ip as-path access-list 1 deny ^?$
!
route-map localonly permit 10
match as-path 1
set local-preference 300
```

For the node R33 the local preference for updates that come from AS4 is set to 300. This value is higher than the local preference value of iBGP updates that come from R3, which value is 200. With this configuration, AS3 set with more preference the node R33 for the local routes from the AS4, any others routes in the node R33 are transmitted internally to the rest of the AS3 with a local preference of 100. Due to this value is lower than 200, the node R3 is the predetermined route. With the access-list is possible to deny any update whose AS_Path attribute starts with a “?” and ends with a “?”, this letter as it was explained before, indicates that has been originated an incomplete route. Such updates that match with the list will be denied. As a result of the filtering the network routing tables are show in the next figures.

***Table 5.6:*** *Result of network filtering*

| Network | Next Hop | Metric | LocPrf | Weight | Path |
|---|---|---|---|---|---|
| *> 40.4.0.0/24 | 1.1.1.1 | 0 | 100 | 32768 | i |
| *>i50.0.0.0/24 | 40.4.0.2 | 0 | 100 | 0 | i |
| *>i90.1.1.0/24 | 40.4.0.2 | 0 | 300 | 0 | i |
| *> 190.12.0.0/24 | 200.1.1.1 | 0 | 200 | 0 | 2 1 5 i |
| *> 193.10.11.0 | 200.1.1.1 | 0 | 200 | 0 | 2 i |
| *> 195.11.14.0 | 200.1.1.1 | 0 | 200 | 0 | 2 1 i |
| *> 200.1.1.0 | 1.1.1.1 | 0 | 200 | 32768 | i |

| Network | Next Hop | Metric | LocPrf | Weight | Path |
|---|---|---|---|---|---|
| *>i40.4.0.0/24 | 50.0.0.2 | 0 | 100 | 0 | i |
| *> 50.0.0.0/24 | 2.2.2.2 | 0 | 100 | 32768 | i |
| *> 90.1.1.0/24 | 40.4.0.2 | 0 | 300 | 32768 | i |
| *>i190.12.0.0/24 | 200.1.1.1 | 0 | 100 | 0 | 2 1 5 i |
| *>i193.10.11.0 | 200.1.1.1 | 0 | 200 | 0 | 2 i |
| *>i195.11.14.0 | 200.1.1.1 | 0 | 100 | 0 | 2 1 i |
| *>i200.1.1.0 | 200.1.1.1 | 0 | 200 | 0 | 2 i |

The above tables show the routing tables of the nodes R3 and R33 after the filtering. As it can be seen the Local Preference for updates has been modified depending on how the routes have been announced.

## 5.5 AS_Path prepending

Another way to make filtering in the network is to change the AS_Path; AS-Path prepending is a way to manipulate the AS-Path attribute of a BGP route. It allows make the AS_Path longer which makes the route less desirable for inbound traffic. AS-Path prepending can be applied to inbound and outbound direction using route-maps. For this case the network has been modified in order to make the tests, the next figure shows the topology of the network and how the data will flow in the network.

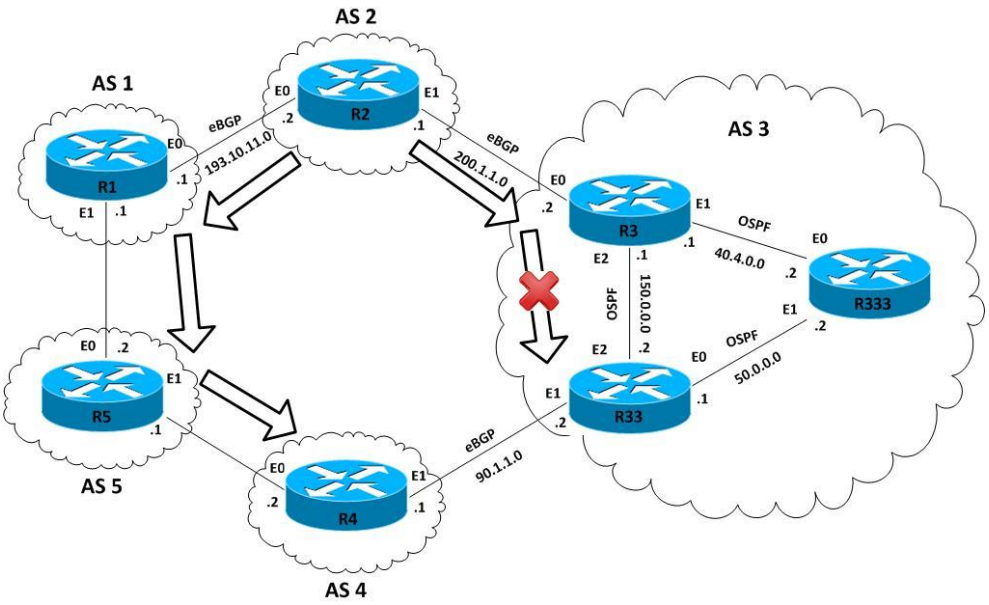



***Figure 5.2:*** *AS_Path Prepending test*

As it can be seen in the figure, when a packet is send to the network 90.1.1.0, originally the packet would follow the shorter path that would be AS3 and then would reach the destination network, but with the command as-path prepend is it possible to modify the path to make it longer and then the packet has to follow different path, in that case, go to the network 90.1.1.0 trough AS1, AS5 and finally AS4. This solution has a drawback; all the traffic of the networks always passes through the same nodes, and this could cause an overload in the network and generates bigger problems. The configuration made for the router R3 in AS3 is the following:

***Table 5.7:*** *AS_Path Prepending configuration*

```
router bgp 3
neighbor 200.1.1.1 remote-as 2
neighbor 200.1.1.1 route-map prepend out
redistribute connected
!
route-map prepend permit 10
set as-path prepend 3 3 3
```

The next tables show the routing table obtained from R4 when wants to request the network 90.1.1.0 and how the AS_Path of the R3 node has been prepended.

***Table 5.8:*** *Results obtained from AS_Path prepending*

```
Router4# show ip bgp
BGP table version is 0, local router ID is 2.2.2.2
   Network          Next Hop            Metric LocPrf Weight Path
*>90.1.1.0/24       193.10.11.1              0             0    1 5 4 i
*                   200.1.11.2               0             0    3 3 3 3 i
```

## 5.6 Other network solutions

In this section will be enumerated the different networks solutions implemented in Netkit that could be a possible solution to solve the problem mentioned in the project.

### 5.6.1 BGP networking without OSPF, full mesh BGP

Another possible solution to avoid the “pollastres de rutes”, how the origin of the problem is the OSPF protocol, in this case is it possible to made a network full meshed with BGP, without using OSPF protocol.

For doing this test, the following network has must set up:

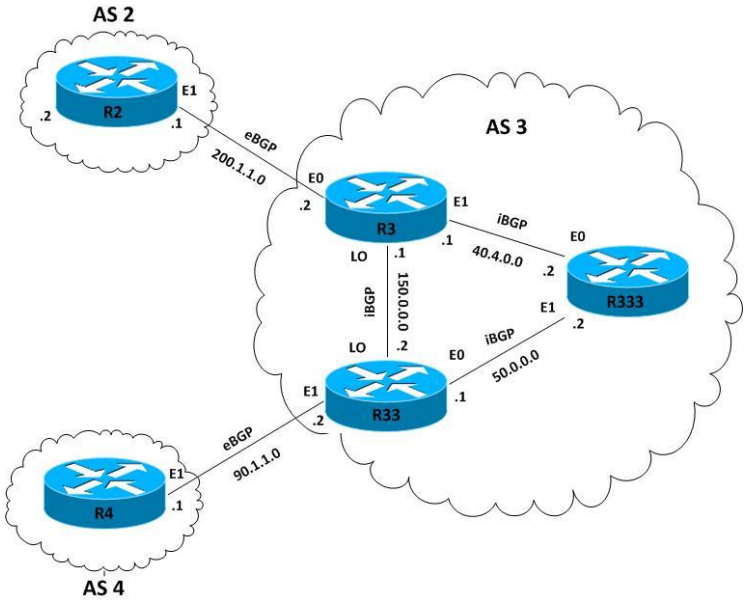



***Figure 5.3:*** *BGP network without OSPF*

As it can be observed, for AS3 it was not used any IGP protocol, only iBGP between the different nodes. For that case is it necessary to modify the configuration files, because synchronization must be turned off to stop BGP speaker from adding routes learned by iBGP, if not could cause a loop in the node and could saturate the node.

In this case all the routers inside the same AS has to know every IP directions of the others node inside the same AS, if not, the routers will never know how to reach the route, because in this case it will never been advertised. The configuration files for this network configuration are placed in the Annex III

As it can be observed in the figure, R2 and R4 are eBGP routers. These routers advertise its own IP address as the next hop to R3 and R333. These next hops does not change in AS3, hence R4 and R2 has already advertises these route to neighbors of the AS. Once this configuration is made, is possible to test the network.
Once the tests are done, it can be conclude that this may not been a possible solution for large networks, because for each node added in the AS, the network administrator should add manually a static route for each of these added nodes, in order to maintain the connectivity, also a single missing neighbor can result in routing loops, also this kind of configuration does not fit with Guifi.net policies. However, this network could be a possible solution to those small networks in which the people who manage the network do not have enough knowledge of networking or even it is not expected a network growth.

## 5.7 BGP Confederation network

After emulating the main causes of generating a “pollastre de rutes”, it has been emulated a final scenario that actually is not implemented in Guifi.net network. However, it was though that this scenario can provide enough information in order to give a possible solution to “pollastres de rutes” that affects in the network by mixing routing protocols.

### 5.7.1 BGP confederation definition

BGP confederation is a method to increase the scalability and improve management in BGP routing policies. The main purpose of this configuration is by subdividing a single AS into multiple sub-AS's. BGP confederation mechanism allows unifying different AS’s under a single AS that will be seen by BGP network globally.

### 5.7.2 BGP confederation configuration

In this test, the network to emulate is similar to the others, but in this case, in the OSPF area will be configured different internal AS’s to configure the BGP confederation.

The advantage that BGP confederation has in front other systems that have been tested, is that with this configuration internal AS’s have a eBGP link between them, then is possible to store the As_Path attribute when the routes are sent over the network. Also another advantage that this network has is that the whole network would see the BGP confederation as a single AS.

The configuration to be done to test the network is the following, in the ASBR of the OSPF zone is it necessary to set additional parameters in order to configure the BGP confederation. Also is it necessary to indicate that between the ASBR there it not direct connection, thus is it necessary to set the parameter “ebgp-multihop” in order to reach the other ASBR node.

***Table 5.9:*** *BGP confederation configuration*

```
router bgp 10
no synchronization
bgp confederation identifier 3
bgp confederation peers 20
network 200.1.1.0/24
network 50.0.0.0/24
neighbor 172.16.0.2 remote-as 20
neighbor 172.16.0.2 ebgp-multihop 2
neighbor 172.16.0.2 update-source 172.16.0.1
neighbor 200.1.1.1 remote-as 2
redistribute connected
redistribute ospf
```

As it can be seen in the above figure, in the OSPF zone each one of the ASBR nodes composes the confederated AS, forming by the confederation of AS3. In next routing tables will be seen how this confederated AS don't send the confederated AS to rest of the network, only between them, and how only is send the AS number of AS3. Also it can be seen in the figure, the eBGP link establish between the ASBR, this will allow to maintain a eBGP connection between the ASBR to maintain the AS_Path attribute in the OSPF zone, this would be better with the information obtained with the routing tables.

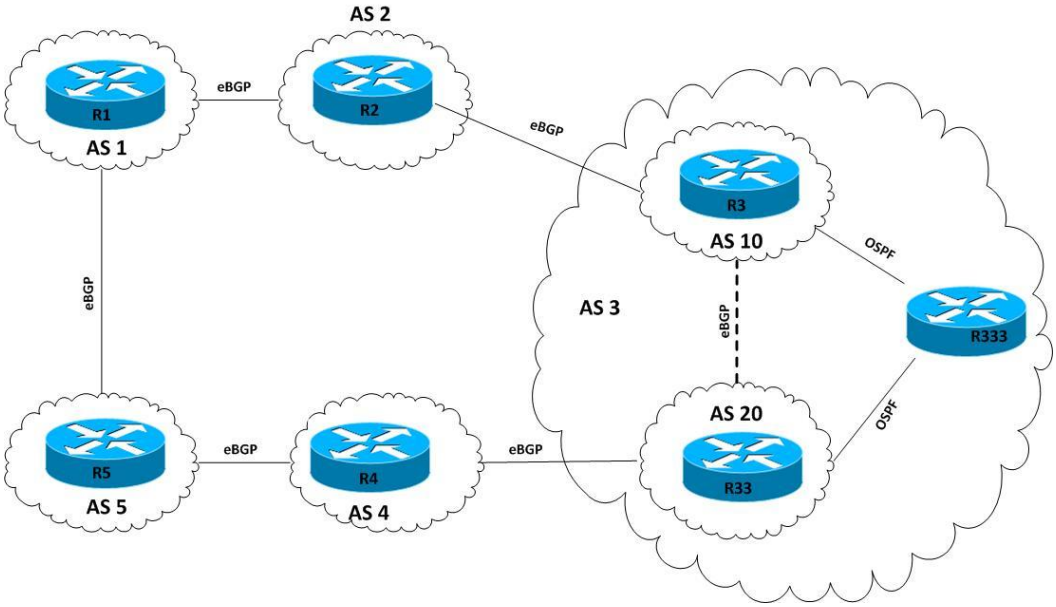



***Figure 5.5:*** *BGP confederation network*

The following routing tables shows the results obtained with the emulation of the BGP confederation network before:

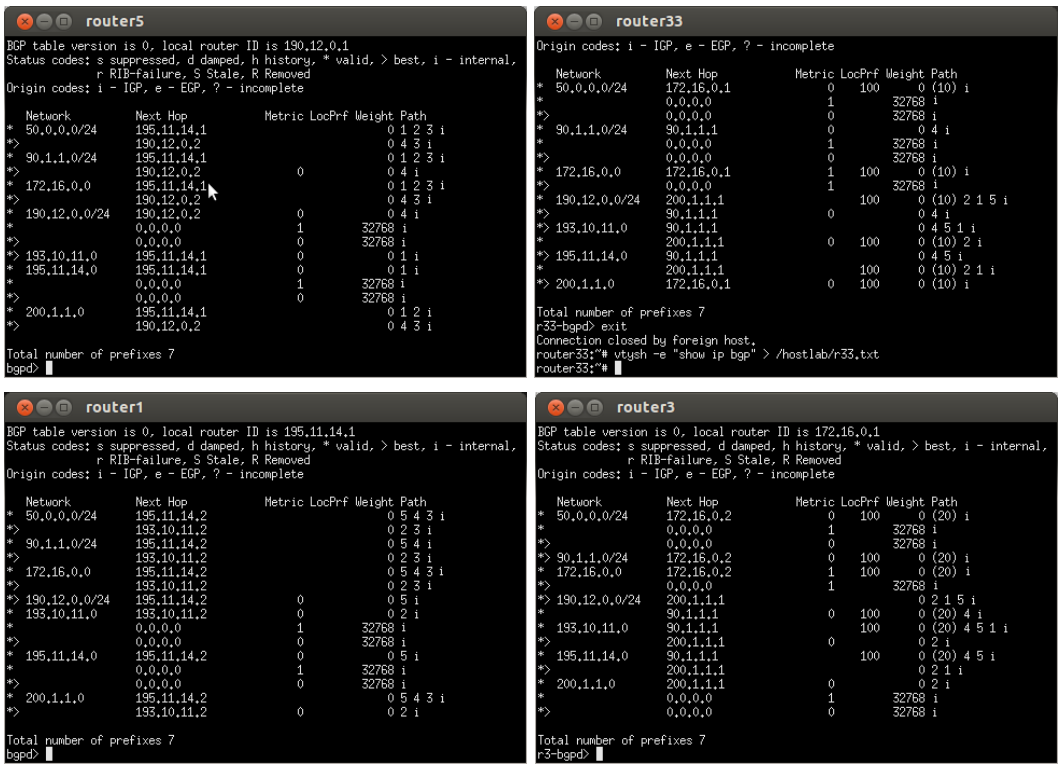



***Figure 5.6:*** *Results of BGP confederation*

### 5.7.3 Conclusions of BGP confederation

Once the network of the previous figure has been emulated, the first thing to notice is the following; external nodes of AS3 do not perceive the confederation configuration made in the ASBR nodes of AS3. Then this confederation is announced to the rest of the network as an only AS3.

Other thing that is possible to observe in the routing tables is for nodes R3 and R33 this confederation exists, appears in the AS_Path between “( )”. This confederation allows creating a BGP link between the two confederates AS and maintaining the AS_Path attribute. With the previous emulated configuration of BGP was no possible to maintain the AS_Path attribute when pass through an OSPF zone. Also is it possible to observe in networks 172.16.0.0 that route incomplete still exist, this route belong to the OSPF zone, this problem can be solved by using filters to avoid this kind of routes.

So, with the results obtained with this network emulation, this scenario probably is a good alternative to the actual Guifi.net network to avoid “pollastres de rutes”, so in the next chapter the idea of this network and the knowledge obtained before will be applied of as an idea of the Guifi.net network.

# CHAPTER 6. IMPLEMENTATION IN GUIFI.NET

## 6.1 Introduction

The main objective of this chapter is design a proposal for Guifi.net network. In this case, is necessary to consider the network as an aggregation of multiple AS which runs an IGP protocol internally and EGP protocol externally to communicate with the other AS's that constitute the network. The protocols and the knowledge used for this purpose are those which have been used throughout the network emulations.

## 6.2 Configuration

To design the network proposal it was decided to choose the zone of Mataró, since in this area there are several AS's connected together by external BGP and in this AS's, internally is running OSPF protocol. Each of these AS is connected between them in order to exchange BGP routing tables, while OSPF is used to assure connection between internal nodes of the AS. The following figure shows the actual configuration of the specified zone:

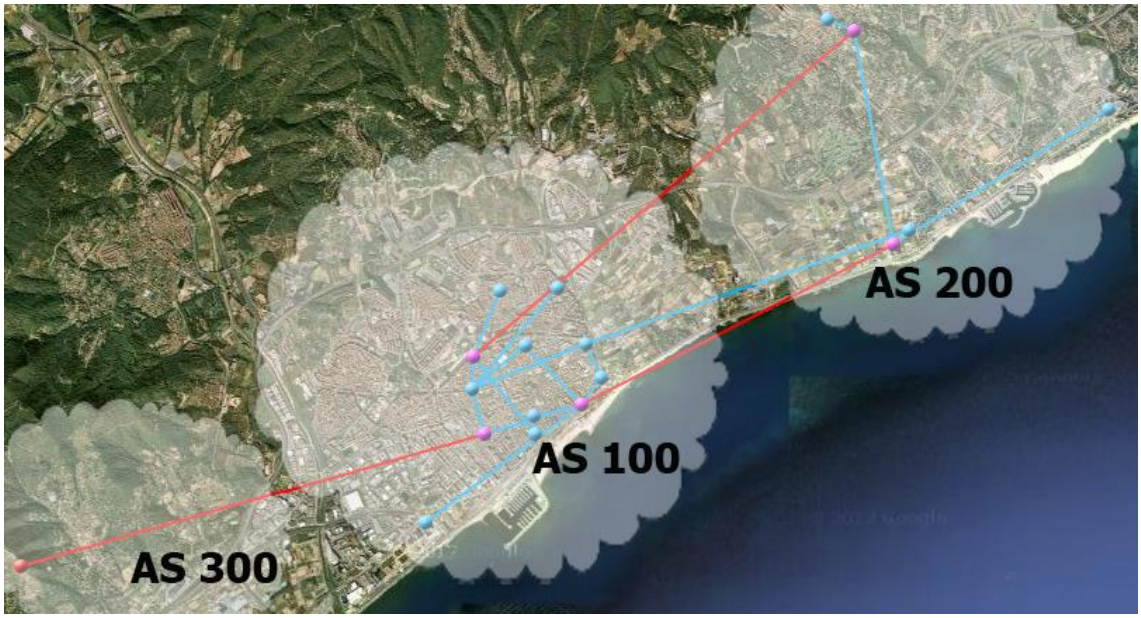



***Figure 6.1:*** *Current configuration of the area selected*

In the figure presented the blue points and lines represent OSPF node, while the red points and lines represent BGP nodes. In case of violet points represents the ASBR nodes of the OSPF zone. As it can be observed in the figure is an aggregation of 3 AS's, AS100 which connects by AS300 and AS200.

In study case presented, OSPF network has communicates with other networks internally by OSPF protocol or externally by BGP protocol. As it was stated before, in case the method chosen to communicate the rest of the network is by BGP, is it necessary to set ASBR in the OSPF zone, which would be the nodes that route the traffic to external AS's and also route traffic inside the same AS. In the previous figure presented, the violet points will be the nodes which interchange information with the external AS's by BGP and also will interchange the information internally by OSPF. But, the inconvenience of doing BGP in an OSPF network, is it necessary to filter the routes carefully, if not the routes would lose the AS_Path attribute. Applying the concepts obtained the following solutions would be deployed in a network as Guifi.net to improve the connection within the different zones. All the configurations must be applied on the boundary nodes of the OSPF/BGP zones.

### 6.2.1 BGP confederation solution

After implementing several scenarios, it has been decided that the model that fits best with Guifi.net is the solution of BGP confederation, since the emulations that have been made have given satisfactory results and in conjunction with other measures such as route filtering and filtering of incomplete path, would be a good solution for network Guifi.net. The figure below shows how would be implemented the configuration of BGP confederation in the selected area.


***Figure 6.2:*** *Configuration of BGP confederation in the selected area*

As it was already explained, in OSPF, the different ASBR nodes that compose the AS, each one forms a confederate AS. Then in this configuration exposed each one of the ASBR node would be configured as a smaller AS which would maintain an eBGP link between each of the boundary nodes, AS10, AS20 and AS30. At the same time these boundary nodes would have established an eBGP link with external AS to assure connectivity to the rest of the network. Also this confederate AS

Hence, by connecting each one of the ASBR's nodes maintaining eBGP link AS_Path attribute would be not lost when routes are exported to external AS's, and also this attribute would be send trough the OSPF zone between the ASBR's. Thus "pollastres de rutes" introduced before, would be avoided as it.

In the case exposed, AS100 is the number of AS that would be exported to the rest of the network, while inside the OSPF area, each one of the ASBR's would exchange their confederate AS number between them in order to assure the connectivity. The confederate AS will not published to the rest of the network.

But even applying this solution that is able to maintain the attribute AS_Path, routing filters should be applied for the exported and imported routes in order to assure better stability of the network. This can be achieved by applying the route filters explained in chapters before and also by filtering the incomplete path coming from the routes of the OSPF area. These filters must be applied in the ASBR, because these nodes are responsible for routing traffic into the OSPF areas and to route traffic to the rest of the network. All these solutions are compatible between them.

### 6.2.2 Load balancing solution

Redundant networks are the routes that have more than one path to connect to external nodes. This presented solution does not solve "pollastres of rutes" but could be a solution for redundant networks in case of a link is broken or a node is crashed.

In a network of the characteristics of Guifi.net, could be interesting to configure this feature, because it is possible to have a default route to route the traffic in one way but in case the node crashes or the link breaks is it possible to have a backup node to route the traffic. Also, with this method would be possible to choose as a default link, one link with more bandwidth than others that connect to external AS's. These decisions are based in the specifications that BGP protocol has.

To perform this action is possible to be done by the BGP command "AS_Path Prepend". This command allows manipulating the AS_Path of a concrete node of an AS in order to make the route longer to external AS's, then the default network chosen would be the route with shorter AS_Path. The next figure shows how the solution of AS_Path prepending could be implemented in Guifi.net network:

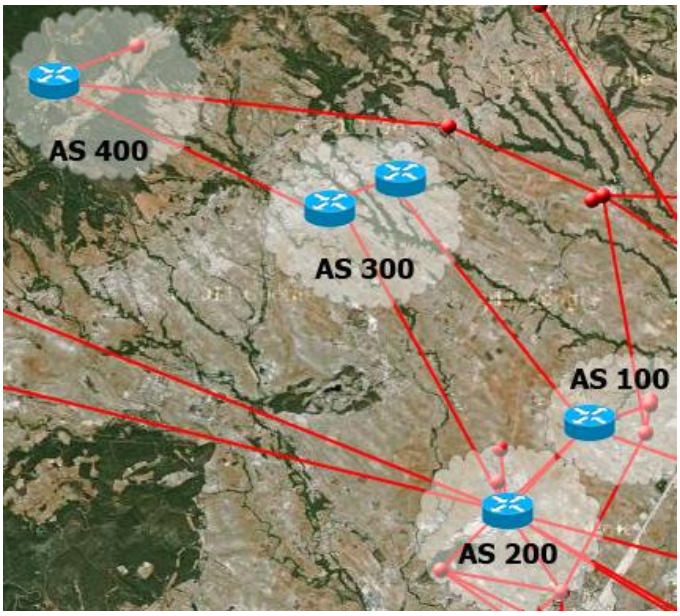



***Figure 6.3:*** *Example of Guifi.net network to use As_Path Prepending*

As it can be observed this solution only could be implemented in zones where more than one path to reach a determined node are, then is it possible to balance the external traffic of the routes through this node.

In the exposed case, would be an example of how AS100 would reach the external AS400. AS100 has two alternatives to reach the AS400, could use the path through AS300 and then reach AS400 or could use the route from AS200, then AS300 and finally AS400. As it can be seen in the figure, the route from AS100 to reach the AS400 is shorter due of the direct link to AS300 and then AS400, but in a network such as Guifi.net, the link that connects this two AS's (AS100 and AS300), could be a link with less capacity or only would be used this link in case of failure the main link (as backup link) of the node of the zone, which is the link of the AS200.

Then, to use the route between AS1000 and AS300 as backup of the main route, the nodes of AS300 directly connected to AS100 must send the route to AS100 using AS_Path prepending command, doing this; AS300 would manipulate the AS_Path making the path longer to AS100. Thus, how AS100 have another alternative to reach AS400 which is by the route of AS200 and this route is shorter than going through AS300, AS100 would chose the AS200 route as default route.

This solution presented also could be complemented with the solution presented above, BGP confederation, offering by this method a greater robustness network.

## 6.3 Conclusion

Assuming that Guifi.net is a network that currently is running despite some specific problems in routing, any implementation that would be performed in the network would not be easy. However, is believed that the result that could be obtained by applying these network improvements would be positive in terms of user experience and better network performance.

For Guifi.net case, the main problem may be dispensed connectivity of some of the areas during the implantation process. But is believed that is possible to configure all BGP nodes that are in the same AS without being altered the operation of the network.

Is it necessary to keep in mind that the whole implementation process involves enough problems to doubt about implement it, since it is a costly process and not always is easy to do and is it necessary to have a plan of action planned.

# CONCLUSIONS

While developing this master thesis has studied the current situation and environment of Guifi.net. The features that distinguish this community network to other similar are the software for the auto configuration of the nodes and the network model followed, which is open and distributed. Free wireless community networks are increasing year after year, so these kinds of communities begin to become a real alternative to existing Internet Service Providers.

Through the project has shown that routing protocols that make up a network like Guifi.net must be differentiated in order to have a stable and optimized network. The scenarios presented in the project are not very realistic in a real network, since only has been emulated a small part of what might be the network. However, by the realization of these emulations have provided enough information to keep in mind that in a network such as Guifi.net, which works under different routing protocols, special attention must be taken with the information propagated of each one of the nodes and AS's that set up the network; because without an adequate control of information may provoke to experience serious problems in the network.

For the realization of the project was necessary to find information about the main causes of failures in the Guifi.net in order to set the objectives of the project, once this was done, it was necessary to emulate each one of these failures to obtain results and propose some of the solutions exposed.

Regarding to these obtained possible solutions for the network, it has been found that route filtering is the most effective method to avoid problems that could appear in the network, but BGP confederation would be an interesting alternative configuration for the network. As has been shown in the emulations this BGP configuration prevents the loss of the AS_Path when the routes passes thought an OSPF zone. Also the implementation of BGP confederations in the network could be complemented with the filtering solution, providing to the users a much more robust network. In addition to give solutions to the problem about mixing OSPF and BGP protocols, by this project has been possible to understand how to balance network traffic and how to apply it in Guifi.net.

Others solutions were implemented as test  like full BGP mesh network, but probably is not a possibility for a community network as Guifi.net, because network administrator has to invest a lot of time adding static routes in the network for each node added in the network and the main objective of the community is to be self-configured.

Regarding to the realization of this project, it has been possible know a Wireless Community Network such as Guifi.net. In this Guifi.net was possible to meet some of the participants who form the community, and exchange opinions with them. These kinds of networks are very important as they provide a service to different kind of users, where anyone can contribute with their knowledge or

skills to improve performance and network usage. Through practical experiences has become aware of the problems present in the design and implementation of small to medium sized community network such as Guifi.net. Also with the realization of this project has been acquired the knowledge of how routing protocols works in Wireless Community Networks. Also was possible to know the main causes of the problems that could appear in a network such as Guifi.net and learn different ways to solve some of these failures.

## Future lines of work

A future line of work will be to improve the system in order to detect when an Update message is sent in the network and execute the script at these time to detect if it is a problem with the routing protocols.

Another possible improvement will be to use different software to data flow; this software could be Pmacct[9], which is set of passive network monitoring tools to measure, account, classify, aggregate and export IPv4 and IPv6 traffic.

Also will be very interesting in a future line of work to test the command “tag”; this command is used in Cisco routers and by this command is possible to retrieve the AS_Path attribute when an OSPF route is redistributed back into BGP protocol.

[9] http://www.pmacct.net/

# BIBLIOGRAPHY

**Routing**
http://rua.ua.es/dspace/bitstream/10045/15586/9/Redes_tema9_sistemas10-11.pdf
http://www.livinginternet.com/i/iw_route_igp_ospf.htm
http://www.9tut.com/ospf-routing-protocol-tutorial
http://www.cisco.com/en/US/tech/tk365/technologies_white_paper09186a00800 94e9e.shtml
http://www.cisco.com/en/US/docs/ios/iproute_bgp/command/reference/irg_bgp3 .pdf
http://www.interlab.ait.ac.th/tein2/Presentations/Basic%20Routing/BGP%20-%20Routing%20Protocol.pdf
http://www.livinginternet.com/i/iw_route_egp_bgp.htm
http://www.livinginternet.com/i/iw_route_egp_egp.htm
http://www.ordenadores-y-portatiles.com/route-reflector.html
http://www.it.uc3m.es/~teldat/Cbra/castellano/protocolos/Dm514v8_Protocolo_ OSPF.PDF
http://www.rfc-es.org/rfc/rfc1219-es.txt

**Guifi.net**
http://www.guifi.net
https://lists.Guifi.net/listinfo/guifi-barcelones
https://lists.Guifi.net/listinfo/guifi-rdes
https://lists.Guifi.net/listinfo/guifi-baixllobregat

**Roma Tre University – NetKit**
http://wiki.netkit.org
http://list.dia.uniroma3.it/pipermail/netkit.users/

**SNMP**
http://www.net-snmp.org/wiki/index.php/Main_Page
http://www.net-snmp.org/support/mailinglists.html
http://www.it-slav.net/blogs/2009/02/05/install-and-configure-snmp-on-ubuntu/

Detecting Incidents in Wireless Mesh Networks using Flow and Routing Information: Lothar Braun, Alexander Klein, Leon Aaron Kaplan, Georg Carle

Taxonomy of IEEE 802.11 Wireless Parameters and Open Source Measurement Tools: Diego Dujovne, Thierry Turletti, Fethi Filali

Designing High Performance Enterprise Wi-Fi Networks: Rohan Murty, Jitendra Padhye, Ranveer Chandra, Alec Wolman, Brian Zill

Usage Patterns in an Urban WiFi Network: Mikhail Afanasyev, Tsuwei Chen, Geoffrey M. Voelker and Alex C. Snoeren

NetReview: Detecting when interdomain routing goes wrong: Andreas Haeberlen, Ioannis Avramopoulos, Jennifer Rexford, and Peter Druschel

B. Fortz. On the evaluation of the reliability of OSPF routing in IP networks. Institut d'Administration et de Gestion.

Tatiana B. Pereira and Lee L. Ling. Network Performance Analysis of an Adaptive OSPF Routing Strategy – Effective Bandwidth Estimation. International Telecommunication Symposium – ITS 2002, Natal, Brazil

T. Bates, R. Chandra, E. Chen. BGP Route Reflection - An Alternative to Full Mesh IBGP. April 2002.


UPC
eetac
Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels
UNIVERSITAT POLITÈCNICA DE CATALUNYA

# ANNEXS

# *ANNEX I: Nekit*

## I.I Architecture of Netkit

The virtual nodes created in Netkit are in reality images Kernel and Linux File System, implemented in UML (User Mode Linux) and connected through each "virtual Hubs" as collision domains.

These collision domains can be connected with the outside world through virtual interfaces "tap" that created in the host kernel, allows a node (or the entire network Netkit virtual) networks connect to "real" to through the host network resources.

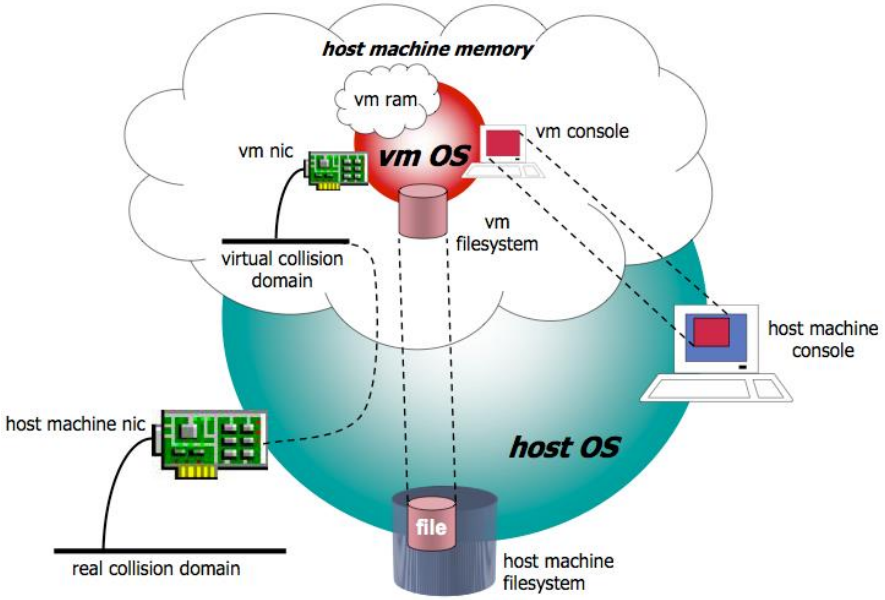



***Figure A1.1:*** *Netkit software structure*

In Netkit software each virtual machine will require the following elements:

- Console (Xterm): through which is accessed the interpreter host system command.
- Memory (Netkit allows to assign a portion of the RAM Host System)
- File system (stored in an image file in the host)
- One or more network interfaces

Packages installed in the file system are available on the Netkit Wiki site.

## I.II How to install Netkit

This is included in the master thesis because is not trivial to install the software and the software will be very useful to people interested in network research.

Netkit only runs under the Linux operating system. The toolkit is fairly independent from of the specific Linux distribution owned.

It is very easy to install Netkit, first of all is it necessary to download 3 principal files to use Netkit. These files are:

- Netkit "core", which contains commands, documentation and other stuff which is necessary for Netkit to work
- Netkit file system, which contains the file system for virtual machines
- Netkit kernel, which contains the kernel used by virtual machines

This is it possible by typing the following commands:

```
# wget http://www.netkit.org/download/netkit/netkit-2.4.tar.bz2

# wget http://www.netkit.org/download/netkit-filesystem/netkit-
filesystem-F2.2.tar.bz2

# wget http://www.netkit.org/download/netkit-kernel/netkit-kernel-
K2.2.tar.bz2
```

Once the files have been downloaded is it necessary to unpack them by using the following commands:

```
Tar –xjSf netkit-x.y.tar.bz2
Tar –xjSf netkit-filesystem-Fx.y.tar.bz2
Tar –xjSf netkit-kernel-Kx.y.tar.bz2
```

It is strongly advised to use the -S option to save space on the hard disk, because this option handles sparse files efficiently.

Once done it the steps before is it necessary to set the following variables to indicate where Netkit it has been installed. Is strongly recommended to add these variables at the end of /etc/profile:

```
NETKIT_HOME=”/usr/share/netkit”
MANPATH=”$MANPATH:$NETKIT_HOME/man”
PATH=”$PATH:$NETKIT_HOME/bin”
Export NETKIT HOME MANPATH PATH
```

It may also be useful to put these lines inside the shell initialization file (`.bashrc' in case bash shell is being used). Consult to shell documentation for more information about this.

At this point, change the current directory to the Netkit directory:

```
Cd netkit
```

Now, run the `check_configuration.sh' script by typing:

```
./check_configuration.sh
```

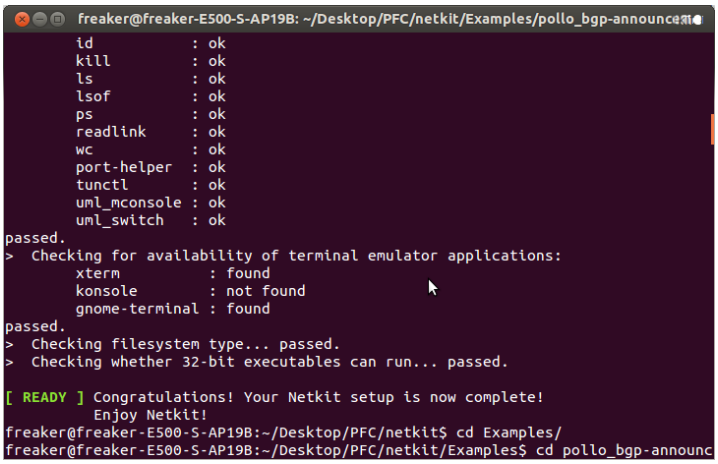



***Figure A1.2:*** *Screen save once Netkit setup is already configured and ready to emulate networks*

This script takes care of checking whether your system is configured properly to make Netkit run. Any misconfigurations are signaled and instructions for fixing them are reported as well. If the script exits with success, then Netkit is ready for use.

Starting a machine

In order to test whether Netkit is working properly is it possible to start a simple virtual machine by typing the command:

```
Vstart pc1
```

If everything is in place, will appear a new virtual machine starting up (eventually popping up an xterm window) and the command `vlist' on the host machine should show an output which is similar to the following:

```
vlist
```

```
USER        VHOST            PID  UPTIME       SIZE  INTERFACES
freaker       pc1            24102  00:03       12376
Total virtual machines:      1    (you),      1    (all users).
Total consumed memory:    12376 KB (you),    12376 KB (all users).
```

Is it possible to stop the virtual machine by typing the following command on the host machine console:

```
Vhalt -r pc1
```

Now is it possible to delete the file pc1.log.

Also is it possible to download some examples that are in the web and configure them. To start a lab downloaded from the web the steps to be done are the following:

Is it necessary to unpack the file, and then in the terminal console where Netkit has been check the configuration, navigate to the directory where the file has been unpacked and then type in the terminal console the command “lstart” that it will start the entire lab as shows the next figure:

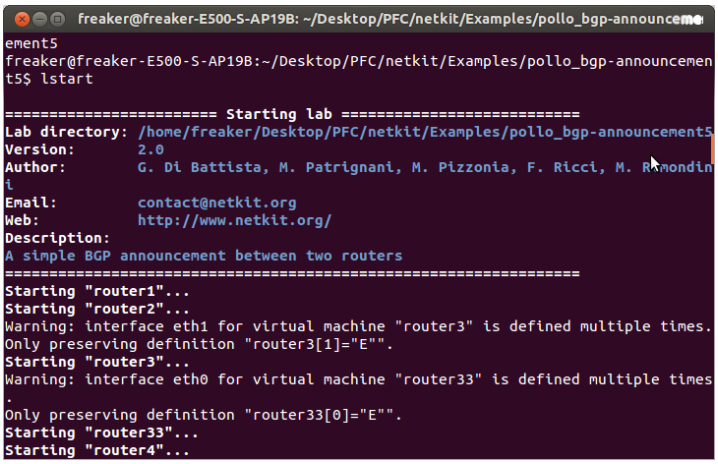



***Figure A1.3:*** *Screen how Netkit is configuring nodes*

### I.III Commands in Netkit

Netkit provides to user two groups of commands:

- V-commands: to start/stop/halt a single machine
- L-commands: to start/stop/halt an entire lab

V-commands

- Vstart: start a virtual machine
- Vlist: list the specifications of the virtual machine that is running
- Vconfig: set the network interfaces that will be used in the virtual machine
- Vhalt: try to stop correctly a virtual machine
- Vcrash: kill the process of a virtual machine
- Vclean: also called “panic command” to clean all the processes and configurations

L-commands

L-commands are the same as V-commands but instead of referring only a single virtual machine refers to the entire lab.

- Lstart: start a virtual lab
- Llist: list the specifications of each virtual machine that there are running

- Lconfig: set the network interfaces that will be used in the virtual machines
- Lhalt: try to stop correctly the entire virtual lab
- Lcrash: kill the process of the virtual lab
- Lclean: clean all the processes and configurations

# *ANNEX II: Initial configurations in the network*

To solve the problems that the network has when routing protocols are mixed in it, has been decided to emulate the network because is not possible to do the test on the real network because can cause several problems to the rest of the network.

As it has been indicated before, to emulate the behavior of the network it has been decided to use the software Netkit.

With the study of the emulate will be able to have a better vision of the problems that occurs when a new node is created, a node crash, a new link is created and other things that could happen in a real network like Guifi.net

## II.I Emulation with BGP

Firstly it has been decided to study only the behavior of a BGP network, with the purpose to once be all already connected, well configured and working, compare this BGP network performance with an OSPF network.

The next figure shows the configuration that has been used to do the BGP tests:

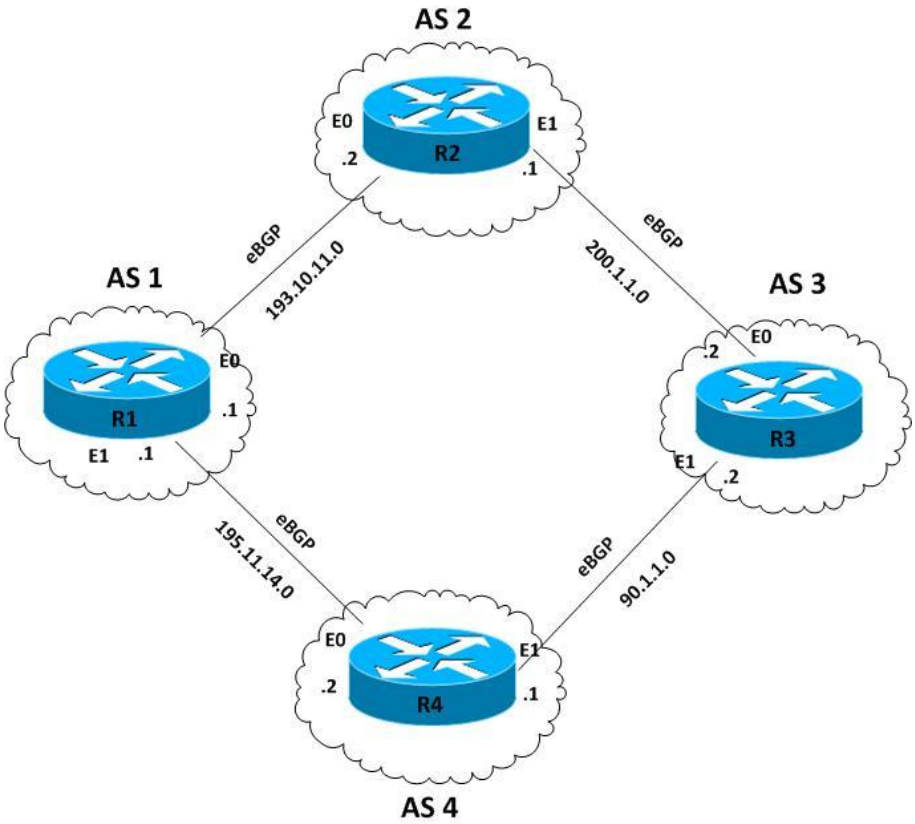



***Figure A2.1:*** *BGP network configuration*

As it can be seen in the figure, the links configured are the following: AS1→ AS2→AS3→AS4→AS1 closing the network and forcing to all the routers to interchange their routing table.

The next figures show the routing tables of each router and how is selected the shortest route with the purpose to connect the destiny router.

Routing table R1

```
BGP table version is 0, local router ID is 195.11.14.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 90.1.1.0/24      195.11.14.2              0             0 4 i
*                    193.10.11.2                           0 2 3 i
*  193.10.11.0      193.10.11.2              0             0 2 i
*>                   0.0.0.0                  0         32768 i
*  195.11.14.0      195.11.14.2              0             0 4 i
*>                   0.0.0.0                  0         32768 i
*  200.1.1.0        195.11.14.2                            0 4 3 i
*>                  193.10.11.2               0             0 2 i

Total number of prefixes 4
```

AS it can be seen in the Routing table of R1 contains all the possible routes that can be in the network and selects the shortest path to one destiny. For example to arrive to the route 90.1.1.0/24 there are two possibilities, go to the next hop in 195.11.14.2 and go across AS4 or choose the hop 193.10.11.2 and go across AS2 and AS3. In this case how the shortest path is the hop 195.11.14.2 is this one which is selected as predetermined.

Routing table R2

```
BGP table version is 0, local router ID is 200.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  90.1.1.0/24      193.10.11.1                            0 1 4 i
*>                   200.1.1.2                0             0 3 i
*  193.10.11.0      193.10.11.1              0             0 1 i
*>                   0.0.0.0                  0         32768 i
*  195.11.14.0      200.1.1.2                              0 3 4 i
*>                   193.10.11.1              0             0 1 i
*  200.1.1.0        200.1.1.2                0             0 3 i
*>                  0.0.0.0                   0         32768 i

Total number of prefixes 4
```

Routing table R3

```
BGP table version is 0, local router ID is 90.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
```

```
          r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  90.1.1.0/24      90.1.1.1                 0             0 4 i
*>                  0.0.0.0                  0         32768 i
*  193.10.11.0      90.1.1.1                               0 4 1 i
*>                  200.1.1.1                0             0 2 i
*> 195.11.14.0      90.1.1.1                 0             0 4 i
*                   200.1.1.1                              0 2 1 i
*  200.1.1.0        200.1.1.1                0             0 2 i
*>                  0.0.0.0                  0         32768 i

Total number of prefixes 4
```

Routing table R4

```
BGP table version is 0, local router ID is 90.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
          r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  90.1.1.0/24      90.1.1.2                 0             0 3 i
*>                  0.0.0.0                  0         32768 i
*> 193.10.11.0      195.11.14.1              0             0 1 i
*                   90.1.1.2                               0 3 2 i
*  195.11.14.0      195.11.14.1              0             0 1 i
*>                  0.0.0.0                  0         32768 i
*  200.1.1.0        195.11.14.1                            0 1 2 i
*>                  90.1.1.2                 0             0 3 i

Total number of prefixes
```

As it can be seen in each routing table, each router receives 2 routes from the other routers, with the possibility of going by two paths to connect to the other AS of the network.

### II.II Emulation with BGP and OSPF

As it was mentioned previously the problem appears when two different routing protocols are mixed. To reproduce this problem, it has been used the previously example but with a modification, in AS3 instead of having only a BGP link there is added another router in the AS where the link between these two routers are made by a OSPF link.

The configuration of the proposed network will be showed in the next figure:

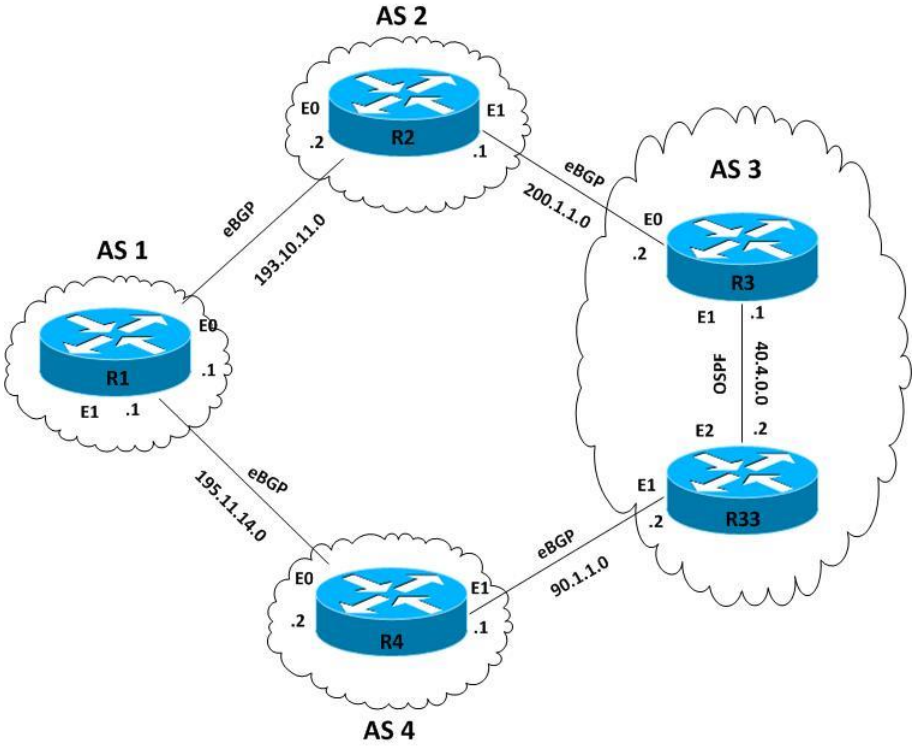



***Figure A2.2:*** *BGP network configuration with AS3 with OSPF as routing protocol*

Running this configuration is possible to see the problems that has including an OSPF link in the network, watching the routing table in R4 is it possible to see the routes originated by AS2. There are two routes originated by AS2, one of them that pass across AS1 and AS2 and the other route that only goes by AS3

The first route is it correct, because as it was explained before in BGP network, but the second route is not ok, because when the network is entirely closed, like in this example, when a packet pass across an OSPF zone loses his AS_Path, with the consequence that the routing table will have and incorrect information because will consider that the information has been generated by the AS3 instead of AS2 and then the router will select as predetermined the route with the shortest path.

AS4 will notify to the rest of the AS the information that has in the BGP routing table and then the other AS will change the information that has in the routing table with a wrong information. Then the entire network will be affected by this wrong information initiated in AS3 making the problem known as “pollastre de rutes”

But making this problem, the network seems that Works ok, because if pings or traceroutes are made the response of the routers will be ok, but the information that contains in the routing table will be wrong.

Now the routing tables will be shown to see better the problem that it has:

Routing table R1

```
BGP table version is 0, local router ID is 195.11.14.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  40.4.0.0/24      195.11.14.2                              0 4 3 i
*>                  193.10.11.2                              0 2 3 i
*> 90.1.1.0/24      195.11.14.2              0               0 4 i
*                   193.10.11.2                              0 2 3 i
*  193.10.11.0      193.10.11.2              0               0 2 i
*>                  0.0.0.0                  0           32768 i
*  195.11.14.0      195.11.14.2              0               0 4 i
*>                  0.0.0.0                  0           32768 i
*  200.1.1.0        195.11.14.2                              0 4 3 i
*>                  193.10.11.2              0               0 2 i

Total number of prefixes 5
```

## Routing table R2

```
BGP table version is 0, local router ID is 200.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 40.4.0.0/24      200.1.1.2                0               0 3 i
*  90.1.1.0/24      193.10.11.1                              0 1 4 i
*>                  200.1.1.2                0               0 3 i
*  193.10.11.0      200.1.1.2                0               0 3 i
*                   193.10.11.1              0               0 1 i
*>                  0.0.0.0                  0           32768 i
*  195.11.14.0      200.1.1.2                0               0 3 i
*>                  193.10.11.1              0               0 1 i
*  200.1.1.0        200.1.1.2                0               0 3 i
*>                  0.0.0.0                  0           32768 i

Total number of prefixes 5
```

## Routing table R3

```
BGP table version is 0, local router ID is 40.4.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 40.4.0.0/24      0.0.0.0                  0           32768 i
*> 90.1.1.0/24      0.0.0.0                  0           32768 i
*  193.10.11.0      200.1.1.1                0               0 2 i
*>                  0.0.0.0                  0           32768 i
*  195.11.14.0      200.1.1.1                               0 2 1 i
*>                  0.0.0.0                  0           32768 i
*  200.1.1.0        200.1.1.1                0               0 2 i
*>                  0.0.0.0                  0           32768 i
```

```
Total number of prefixes 5
```

## Routing table R33

```
BGP table version is 0, local router ID is 90.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 40.4.0.0/24      0.0.0.0                  0         32768 i
*  90.1.1.0/24      90.1.1.1                 0             0 4 i
*>                  0.0.0.0                  0         32768 i
*  193.10.11.0      90.1.1.1                               0 4 1 i
*>                  0.0.0.0                  0         32768 i
*  195.11.14.0      90.1.1.1                 0             0 4 i
*>                  0.0.0.0                  0         32768 i
*> 200.1.1.0        0.0.0.0                  0         32768 i

Total number of prefixes 5
```

## Routing table R4

```
BGP table version is 0, local router ID is 90.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 40.4.0.0/24      90.1.1.2                 0             0 3 i
*                   195.11.14.1                            0 1 2 3 i
*  90.1.1.0/24      90.1.1.2                 0             0 3 i
*>                  0.0.0.0                  0         32768 i
*  193.10.11.0      90.1.1.2                 0             0 3 i
*>                  195.11.14.1              0             0 1 i
*  195.11.14.0      90.1.1.2                 0             0 3 i
*                   195.11.14.1              0             0 1 i
*>                  0.0.0.0                  0         32768 i
*> 200.1.1.0        90.1.1.2                 0             0 3 i
*                   195.11.14.1                            0 1 2 i

Total number of prefixes 5
```

As it can be seen in this last routing table, it can be observed the problem explained before, where the routes that passes across AS3 only has as path AS3 and don’t have any other route to other destinies.

# *ANNEX III: Configuration files for the tests*

## III.I BGP OSPF network configuration

```
LAB_DESCRIPTION="A BGP network working with OSPF"
LAB_VERSION=2.0
LAB_AUTHOR="A. Andrés"
LAB_EMAIL=a.andres.rodriguez@gmail.com
LAB_WEB=http://www.netkit.org/

router1[0]="A"
router2[0]="A"

router1[1]="B"
router5[0]="B"

router3[0]="C"
router2[1]="C"

router3[1]="E"
router333[0]="E"

router3[2]="H"
router33[0]="H"

router33[2]="G"
router333[2]="G"

router33[1]="D"
router4[1]="D"

router4[0]="F"
router5[1]="F"
```

```
R1
/sbin/ifconfig eth0 193.10.11.1 netmask 255.255.255.0 up
/sbin/ifconfig eth1 195.11.14.1 netmask 255.255.255.0 up
/etc/init.d/zebra start
R2
/sbin/ifconfig eth0 193.10.11.2 netmask 255.255.255.0 up
/sbin/ifconfig eth1 200.1.1.1 netmask 255.255.255.0 up
/etc/init.d/zebra start
R3
/sbin/ifconfig eth0 200.1.1.2 netmask 255.255.255.0 up
/sbin/ifconfig eth1 40.4.0.1 netmask 255.255.255.0 up
/sbin/ifconfig eth2 150.0.0.1 netmask 255.255.255.0 down
ifconfig lo:1 1.1.1.1 netmask 255.255.255.255 up
/etc/init.d/zebra start
R4
/sbin/ifconfig eth0 190.12.0.2 netmask 255.255.255.0 up
/sbin/ifconfig eth1 90.1.1.1 netmask 255.255.255.0 up
/etc/init.d/zebra start
R5
/sbin/ifconfig eth0 195.11.14.2 netmask 255.255.255.0 up
/sbin/ifconfig eth1 190.12.0.1 netmask 255.255.255.0 up
```

```
/etc/init.d/zebra start
R33
/sbin/ifconfig eth0 150.0.0.2 netmask 255.255.255.0 down
/sbin/ifconfig eth1 90.1.1.2 netmask 255.255.255.0 up
/sbin/ifconfig eth2 50.0.0.1 netmask 255.255.255.0 up
ifconfig lo:1 2.2.2.2 netmask 255.255.255.255 up
/etc/init.d/zebra start
R333
/sbin/ifconfig eth0 40.4.0.2 netmask 255.255.255.0 up
/sbin/ifconfig eth2 50.0.0.2 netmask 255.255.255.0 up
/etc/init.d/zebra start
```

```
router bgp 1
network 195.11.14.0/24
network 193.10.11.0/24
neighbor 193.10.11.2 remote-as 2
neighbor 193.10.11.2 description (Virtual) Router 2 of AS2
neighbor 195.11.14.2 remote-as 5
neighbor 195.11.14.2 description (Virtual) Router 5 of AS5
redistribute connected
```

```
router bgp 2
network 200.1.1.0/24
network 193.10.11.0/24
neighbor 193.10.11.1 remote-as 1
neighbor 193.10.11.1 description (Virtual) Router 1 of AS1
neighbor 200.1.1.2 remote-as 3
neighbor 200.1.1.2 description (Virtual) Router 3 of AS3
redistribute connected
```

```
router bgp 4
network 190.12.0.0/24
network 90.1.1.0/24
neighbor 190.12.0.1 remote-as 5
neighbor 190.12.0.1 description (Virtual) Router 5 of AS5
neighbor 90.1.1.2 remote-as 3
neighbor 90.1.1.2 description (Virtual) Router 3 of AS3
redistribute connected
```

```
router bgp 5
network 195.11.14.0/24
network 190.12.0.0/24
neighbor 195.11.14.1 remote-as 1
neighbor 195.11.14.1 description (Virtual) Router 1 of AS1
neighbor 190.12.0.2 remote-as 4
neighbor 190.12.0.2 description (Virtual) Router 4 of AS4
redistribute connected
```

```
router bgp 3
no synchronization
network 200.1.1.0/24
network 40.4.0.0/24
network 150.0.0.0/24
neighbor 2.2.2.2 remote-as 3
neighbor 2.2.2.2 update-source 1.1.1.1
```

```
neighbor 2.2.2.2 description r33
neighbor 200.1.1.1 remote-as 2
neighbor 200.1.1.1 description (Virtual) Router 2 of AS2
redistribute connected
redistribute ospf

router ospf
passive-interface eth0
network 40.4.0.0/24 area 0.0.0.0
!network 150.0.0.0/24 area 0.0.0.0
redistribute connected
redistribute bgp
```

```
router bgp 3
no synchronization
network 90.1.1.0/24
network 150.0.0.0/24
network 50.0.0.0/24
neighbor 1.1.1.1 remote-as 3
neighbor 1.1.1.1 update-source 2.2.2.2
neighbor 1.1.1.1 description r3
neighbor 90.1.1.1 remote-as 4
neighbor 90.1.1.1 description (Virtual) Router 4 of AS4
redistribute connected
redistribute ospf

router ospf
passive-interface eth1
network 50.0.0.0/24 area 0.0.0.0
!network 150.0.0.0/24 area 0.0.0.0
redistribute connected
redistribute bgp
```

```
router ospf
network 40.4.0.0/24 area 0.0.0.0
network 50.0.0.0/24 area 0.0.0.0
redistribute connected
redistribute bgp
```

## III.II Prefix filtering configuration

```
router bgp 1
network 195.11.14.0/24
network 193.10.11.0/24
neighbor 193.10.11.2 remote-as 2
neighbor 193.10.11.2 description (Virtual) Router 2 of AS2
neighbor 195.11.14.2 remote-as 5
neighbor 195.11.14.2 description (Virtual) Router 5 of AS5
redistribute connected
```

```
router bgp 2
network 200.1.1.0/24
network 193.10.11.0/24
neighbor 193.10.11.1 remote-as 1
```

```
neighbor 193.10.11.1 description (Virtual) Router 1 of AS1
neighbor 200.1.1.2 remote-as 3
neighbor 200.1.1.2 description (Virtual) Router 3 of AS3
redistribute connected
```

```
router bgp 4
network 190.12.0.0/24
network 90.1.1.0/24
neighbor 190.12.0.1 remote-as 5
neighbor 190.12.0.1 description (Virtual) Router 5 of AS5
neighbor 90.1.1.2 remote-as 3
neighbor 90.1.1.2 description (Virtual) Router 3 of AS3
redistribute connected
```

```
router bgp 5
network 195.11.14.0/24
network 190.12.0.0/24
neighbor 195.11.14.1 remote-as 1
neighbor 195.11.14.1 description (Virtual) Router 1 of AS1
neighbor 190.12.0.2 remote-as 4
neighbor 190.12.0.2 description (Virtual) Router 4 of AS4
redistribute connected
```

```
router bgp 3
no synchronization
network 90.1.1.0/24
network 150.0.0.0/24
network 50.0.0.0/24
neighbor 1.1.1.1 remote-as 3
neighbor 1.1.1.1 update-source 2.2.2.2
neighbor 1.1.1.1 description r3
neighbor 90.1.1.1 remote-as 4
neighbor 90.1.1.1 description (Virtual) Router 4 of AS4
neighbor 90.1.1.1 distribute-list 1 out
redistribute connected
redistribute ospf
!
access-list 1 deny 50.0.0.0 255.255.255.0
access-list 1 permit 0.0.0.0 255.255.255.255
```

```
router bgp 3
no synchronization
network 200.1.1.0/24
network 40.4.0.0/24
network 150.0.0.0/24
neighbor 2.2.2.2 remote-as 3
neighbor 2.2.2.2 update-source 1.1.1.1
neighbor 2.2.2.2 description r33
neighbor 200.1.1.1 remote-as 2
neighbor 200.1.1.1 description (Virtual) Router 2 of AS2
neighbor 200.1.1.1 prefix-list partial out
redistribute connected
redistribute ospf
!
```

```
distribute-list 1 deny 40.4.0.0 255.255.255.0
distribute-list 1 permit 0.0.0.0 255.255.255.255
```

### III.III AS_Path prepending configuration

```
router bgp 1
network 195.11.14.0/24
network 193.10.11.0/24
neighbor 193.10.11.2 remote-as 2
neighbor 193.10.11.2 description (Virtual) Router 2 of AS2
neighbor 195.11.14.2 remote-as 5
neighbor 195.11.14.2 description (Virtual) Router 5 of AS5
redistribute connected
```

```
router bgp 2
network 200.1.1.0/24
network 193.10.11.0/24
neighbor 193.10.11.1 remote-as 1
neighbor 193.10.11.1 description (Virtual) Router 1 of AS1
neighbor 200.1.1.2 remote-as 3
neighbor 200.1.1.2 description (Virtual) Router 3 of AS3
redistribute connected
```

```
router bgp 4
network 190.12.0.0/24
network 90.1.1.0/24
neighbor 190.12.0.1 remote-as 5
neighbor 190.12.0.1 description (Virtual) Router 5 of AS5
neighbor 90.1.1.2 remote-as 3
neighbor 90.1.1.2 description (Virtual) Router 3 of AS3
redistribute connected
```

```
router bgp 5
network 195.11.14.0/24
network 190.12.0.0/24
neighbor 195.11.14.1 remote-as 1
neighbor 195.11.14.1 description (Virtual) Router 1 of AS1
neighbor 190.12.0.2 remote-as 4
neighbor 190.12.0.2 description (Virtual) Router 4 of AS4
redistribute connected
```

```
router bgp 3
no synchronization
network 90.1.1.0/24
network 150.0.0.0/24
network 50.0.0.0/24
neighbor 1.1.1.1 remote-as 3
neighbor 1.1.1.1 update-source 2.2.2.2
neighbor 1.1.1.1 description r3
neighbor 90.1.1.1 remote-as 4
neighbor 90.1.1.1 description (Virtual) Router 4 of AS4
neighbor 90.1.1.1 route-map prepend out
redistribute connected
```

```
redistribute ospf
!
route-map prepend permit 10
set as-path prepend 3 3
```

```
router bgp 3
no synchronization
network 200.1.1.0/24
network 40.4.0.0/24
network 150.0.0.0/24
neighbor 2.2.2.2 remote-as 3
neighbor 2.2.2.2 update-source 1.1.1.1
neighbor 2.2.2.2 description r33
neighbor 200.1.1.1 remote-as 2
neighbor 200.1.1.1 description (Virtual) Router 2 of AS2
redistribute connected
redistribute ospf
```

## III.IV Full Mesh network configuration

```
hostname r3-bgpd
router bgp 3
no synchronization
network 40.4.0.0 mask 255.255.255.0
network 150.0.0.0 mask 255.255.255.0
network 200.1.1.0 mask 255.255.255.0
neighbor 150.0.0.2 remote-as 3
neighbor 50.0.0.1 remote-as 3
neighbor 50.0.0.2 remote-as 3
neighbor 40.4.0.2 remote-as 3
neighbor 200.1.1.1 remote-as 2
no auto-summary
```

```
hostname r33-bgpd
router bgp 3
no synchronization
network 50.0.0.0 mask 255.255.255.0
network 150.0.0.0 mask 255.255.255.0
network 90.1.1.0 mask 255.255.255.0
neighbor 150.0.0.1 remote-as 3
neighbor 50.0.0.2 remote-as 3
neighbor 40.4.0.1 remote-as 3
neighbor 40.4.0.2 remote-as 3
neighbor 90.1.1.1 remote-as 4
no auto-summary
```

```
hostname r333-bgpd
router bgp 3
no synchronization
network 40.4.0.0 mask 255.255.255.0
network 150.0.0.0 mask 255.255.255.0
neighbor 150.0.0.1 remote-as 3
neighbor 150.0.0.2 remote-as 3
```

```
neighbor 50.0.0.1 remote-as 3
neighbor 40.4.0.1 remote-as 3
no auto-summary
```

```
hostname r4-bgpd
router bgp 4
no synchronization
network 90.1.1.0 mask 255.255.255.0
neighbor 90.1.1.2 remote-as 3
no auto-summary
```

```
hostname r2-bgpd
router bgp 2
no synchronization
network 200.1.1.0 mask 255.255.255.0
neighbor 200.1.1.2 remote-as 3
no auto-summary
```

## III.V BGP confederation configuration

```
LAB_DESCRIPTION="A BGP confederation network working with OSPF"
LAB_VERSION=2.0
LAB_AUTHOR="A. Andrés"
LAB_EMAIL=a.andres.rodriguez@gmail.com
LAB_WEB=http://www.netkit.org/

router1[0]="A"
router2[0]="A"

router1[1]="B"
router5[0]="B"

router3[0]="C"
router2[1]="C"

router3[1]="E"
router333[0]="E"

router3[2]="H"
router33[0]="H"

router33[2]="G"
router333[2]="G"

router33[1]="D"
router4[1]="D"

router4[0]="F"
router5[1]="F"
```

```
router bgp 1
network 195.11.14.0/24
network 193.10.11.0/24
neighbor 193.10.11.2 remote-as 2
```

```
neighbor 193.10.11.2 description (Virtual) Router 2 of AS2
neighbor 195.11.14.2 remote-as 5
neighbor 195.11.14.2 description (Virtual) Router 5 of AS5
redistribute connected
```

```
router bgp 2
network 200.1.1.0/24
network 193.10.11.0/24
neighbor 193.10.11.1 remote-as 1
neighbor 193.10.11.1 description (Virtual) Router 1 of AS1
neighbor 200.1.1.2 remote-as 3
neighbor 200.1.1.2 description (Virtual) Router 3 of AS3
redistribute connected
```

```
router bgp 4
network 190.12.0.0/24
network 90.1.1.0/24
neighbor 190.12.0.1 remote-as 5
neighbor 190.12.0.1 description (Virtual) Router 5 of AS5
neighbor 90.1.1.2 remote-as 3
neighbor 90.1.1.2 description (Virtual) Router 3 of AS3
redistribute connected
```

```
router bgp 5
network 195.11.14.0/24
network 190.12.0.0/24
neighbor 195.11.14.1 remote-as 1
neighbor 195.11.14.1 description (Virtual) Router 1 of AS1
neighbor 190.12.0.2 remote-as 4
neighbor 190.12.0.2 description (Virtual) Router 4 of AS4
redistribute connected
```

```
router bgp 10
no synchronization
bgp confederation identifier 3
bgp confederation peers 20
network 200.1.1.0/24
network 50.0.0.0/24
neighbor 172.16.0.2 remote-as 20
neighbor 172.16.0.2 ebgp-multihop 2
neighbor 172.16.0.2 update-source 172.16.0.1
neighbor 200.1.1.1 remote-as 2
redistribute connected
redistribute ospf

router ospf
passive-interface eth0
network 50.0.0.0/24 area 0.0.0.0
redistribute connected
redistribute bgp
```

```
router bgp 20
no synchronization
bgp confederation identifier 3
bgp confederation peers 10
```

```
network 90.1.1.0/24
network 50.0.0.0/24
neighbor 172.16.0.1 remote-as 10
neighbor 172.16.0.1 ebgp-multihop 2
neighbor 172.16.0.1 update-source 172.16.0.2
neighbor 90.1.1.1 remote-as 4
redistribute connected
redistribute ospf

router ospf
passive-interface eth1
network 50.0.0.0/24 area 0.0.0.0
redistribute connected
redistribute bgp
```

# *ANNEX IV: SNMP*

## IV.I Introduction

The SNMP (Simple Network Management Protocol) is a protocol from the application layer that facilitates the exchange of management information between network devices. Allows administrators to monitor network performance, find and solve problems and plan the network growth.

A network managed by SNMP, consists of three key components: managed devices, agents, network administrators and systems or NMS (Network Management Systems).

Managed devices are those containing an SNMP agent and are within a managed network. Agents are software modules that reside network management in a managed device, are the ones who are responsible for collecting device management information and translate it into a format compatible with SNMP.

## IV.II Installing SNMP

In the tested network is necessary to analyze each of the devices in the network. This will require the installation and configuration of a machine-SNMP agent. In the case developed only is needed to install SNMP on the main machine in which tests are running, because in Netkit software, the SNMP agent is already installed by default only is needed to run it.

The installation of the software itself is simple and is resolved in a single command line:

```
sudo apt-get install snmpd
```

Once the package is installed, is needed to configure the agent, and then summarizes the simple steps for this task:

Go to /etc/snmp/snmpd.conf and edit the file. It is highly recommended to made a backup of it

```
cp /etc/snmp/snmpd.conf /etc/snmp/snmpd.conf.orig
```

With the copy made, is it possible to edit snmpd.conf file. Is it necessary to indicate the following parameters:

```
rocommunity public
syslocation "LH"
syscontact mymail@gmail.com
```

The next step is to make the snmpd use this new file and listen to all interfaces because by default only listen on loopback interface. To do this, is it necessary to edit the file which is stored in /etc/default/snmpd. Is it necessary to change the following parameters:

```
# snmpd options (use syslog, close stdin/out/err).
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -I -smux –p /var/run/snmpd.pid
127.0.0.1'
```

For this configuration

```
# snmpd options (use syslog, close stdin/out/err).
#SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -I -smux -p /var/run/snmpd.pid
127.0.0.1'
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -I -smux -p /var/run/snmpd.pid -
c /etc/
snmp/snmpd.conf'
```

Finally it is important that after configuring the service reboot the SNMP:

```
/etc/init.d/snmpd restart
```

Then is it possible to test the service by the following command:

```
snmpwalk -v 1 –c public –O @ipdestiny
```

```
import netsnmp

def getnetwork():
    oid = netsnmp.VarList(netsnmp.Varbind('.1.3.6.1.2.1.4.21'))

    res = netsnmp.snmpwalk(oid, Version = 2c,
DestHost='192.168.0.100')
    return res

print getnetwork()
```

### IV.III How to execute the script automatically

Also is it possible to configure the script before in order to be executed on the network from time to time, this is possible thanks to the Linux command “cron”.

Running jobs at regular intervals is managed by the command ”cron”, which consists of the “crond daemon” and a set of tables describing what work is to be done and with what frequency. The daemon wakes up every minute and checks the crontabs to determine what needs to be done.

To configure this Linux command is it necessary to write another script into the shell of Linux where is it necessary to specify the following parameters:

1. Minute ( range from 0-59)
2. Hour (range from 0-23)
3. Day of the month (range from 1-31)
4. Month of the year (range from 1-12)
5. Day of the week (range from 0-6, 0 being Sunday)
6. String to be executed by sh (The sixth field is everything after the fifth field, and is interpreted as a string to pass to `sh`)

The following line shows a configuration example of how to configure this:

```
0,20,40 22-23 * 2 fri-sat /home/freaker/test.sh
```

In this example, the command is executed every 20 minutes, for the hours between 10 P.M. and midnight on Fridays and Saturdays during February.

In this case it was no possible to automate the system due of a lack of time and the no possibility to determine when an Update is done in the network. To automate the system could be a future line of work.