Bgp At Doors of Autonomous Systems is Simple (BADASS)

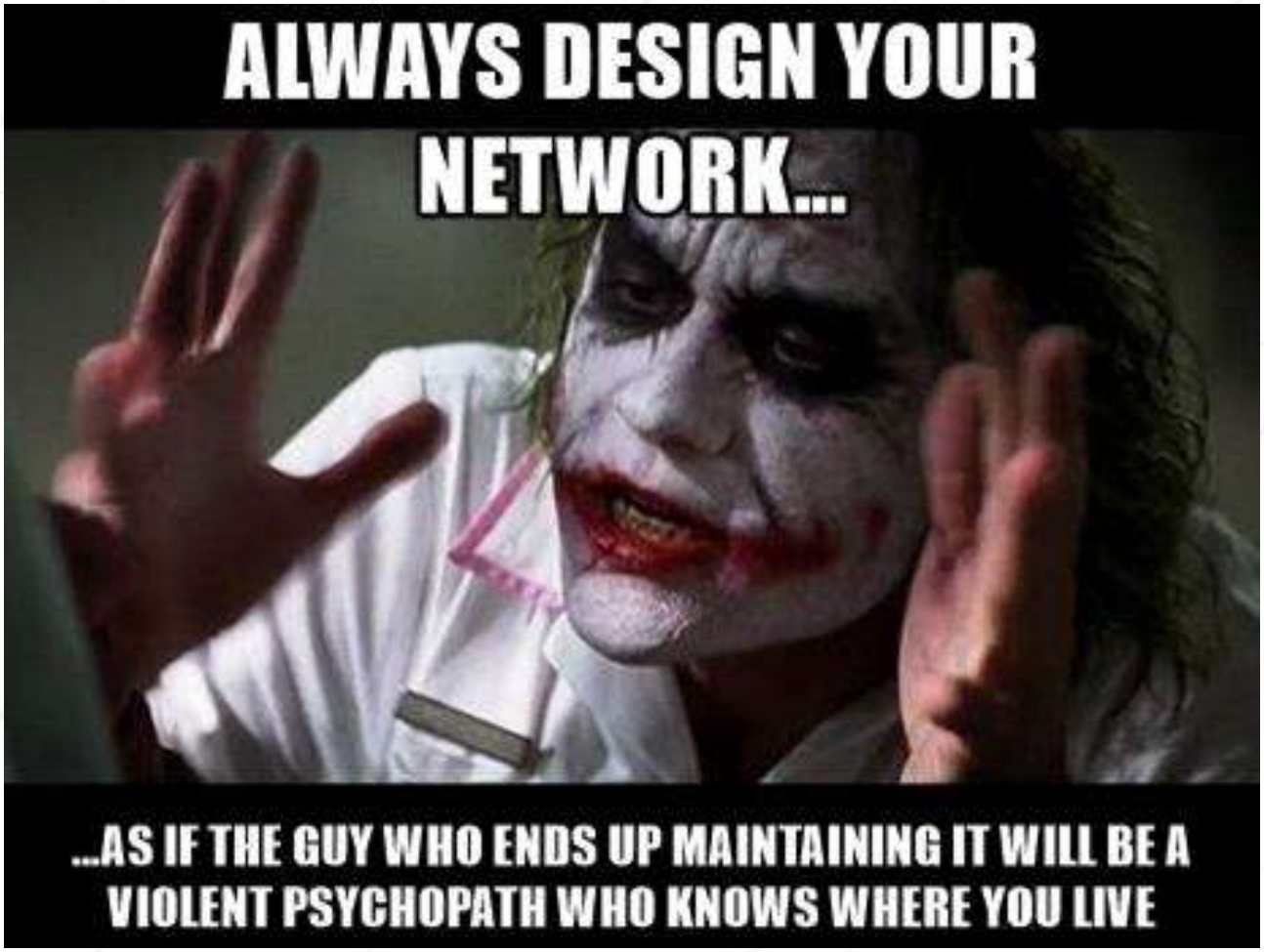
Summary: This document is a subject on network administration.

Version: 2

Contents

Chapter I

Preamble



Chapter II

Introduction

The purpose of this project is to expand on the knowledge you have gained through Net-Practice. You will have to simulate a network and configure it using `GNS3` with `docker` images .

BGP EVPN is based on BGP (RFC 4271) and its extensions MP-BGP (RFC4760). BGP is the routing protocol that drives the Internet. Through MP-BGP extensions, it can be used to carry reachability information (NLRI) for various protocols (IPv4, IPv6, L3 VPN and in this case, EVPN). EVPN is a special family for publishing information about MAC addresses and the end devices that access them..

Chapter III

General guidelines

- The whole project has to be done in a **virtual machine**.
- This project involves using and installing `docker` as well as `GNS3`.
- You have to put all the configuration files of your project in folders located at the root of your repository (go to Submission and peer-evaluation for more information). The folders of the mandatory part will be named: `P1`, `P2` and `P3`.

`WARNING: This project uses many new concepts. Don't be afraid to take some time to read up on how BGP and VXLANs work. Many new terms are voluntarily used.`

`WARNING: This subject contains images, they serve as examples and may contain typography that does not reflect what you should render.`

Chapter IV

Mandatory part

This project will consist of setting up several environments under specific rules.

It is divided into three parts you have to do in the following order:

- Part 1: GNS3 configuration with Docker.
- Part 2: Discovering a VXLAN.
- Part 3: Discovering BGP with EVPN.

You must read the **whole subject** to understand what you need to do.

IV.1 Part 1: GNS3 configuration with Docker

For this first part you will have to configure `GNS3`. It is thus necessary to install and configure `GNS3` as well as `docker` in your virtual machine.

Now that everything works you need to use two `docker` images that you have to make.

A first image with a system of your choice containing at least `busybox` or an equivalent solution.

`Alpine` seems to be a good solution.

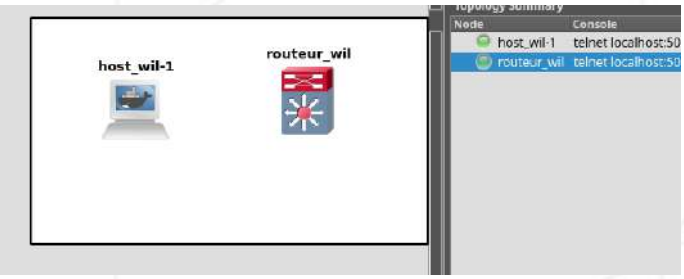
A second image using a system of your choice with the following constraints:

- A software that manages packet routing (zebra or quagga)..
- The service `BGPD` active and configured.
- The service `OSPFD` active and configured.
- An IS-IS routing engine service.
- `Busybox` or an equivalent.

There are pre-built images that need to be configured with this kind of service. Your containers must work in GNS3 with the requested services. You can add what you wish to realize this project

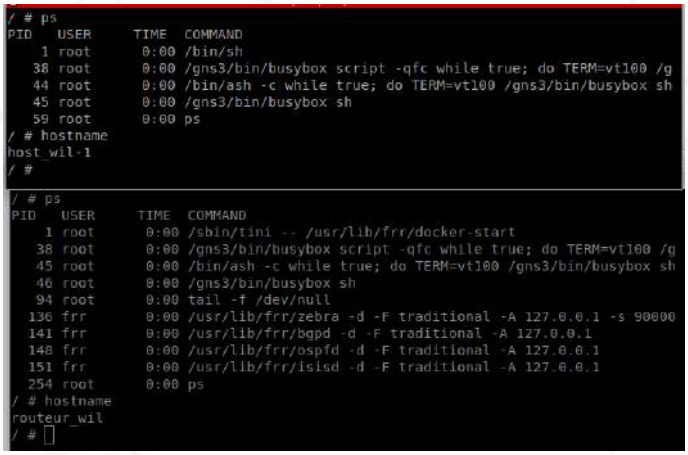
Warning: Your images will be used throughout this project. No IP address should be configured by default.

You must use these two `docker` images in `GNS3` and realize this small diagram. You need to have both machines working. We must be able to connect to them by `GNS3`.



The name of the machines is not put at random it will be necessary to have your login in the name of each equipment (here wil).

Below is an example of each image configured in `GNS3`:

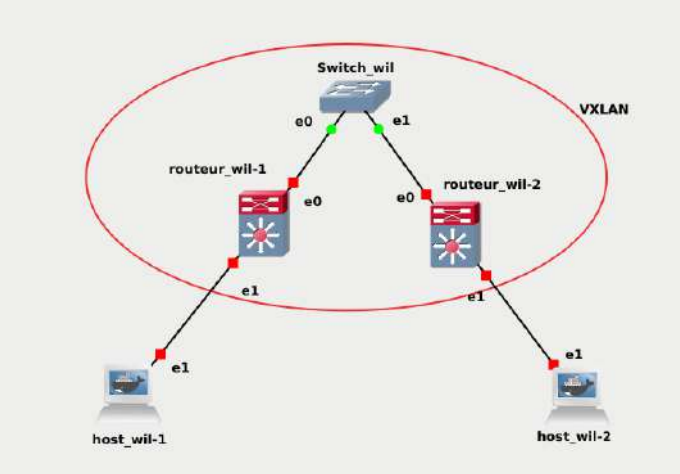


You must render this project in a `P1` folder at the root of your git repository. You should also add the configuration files with comments to explain the set up of each equipment.

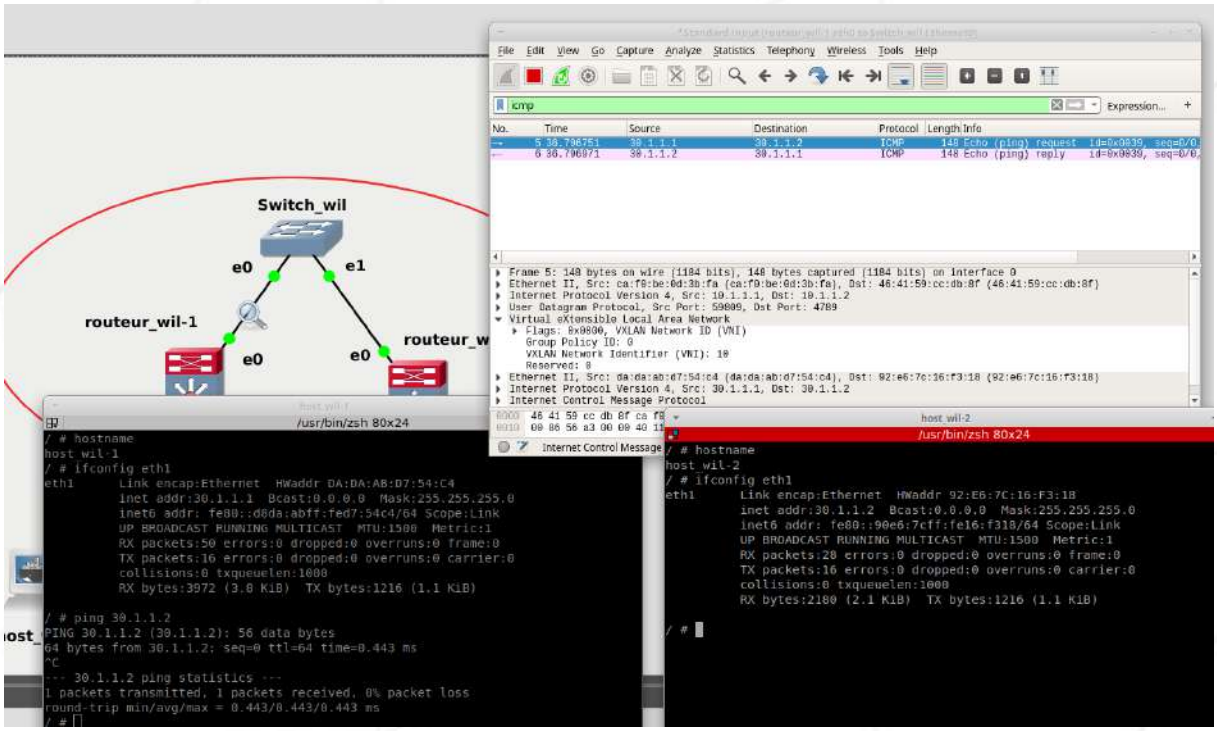
You must export this project with a ZIP compression including the base images. This file must be visible in your git repository

IV.2 Part 2: Discovering a VXLAN

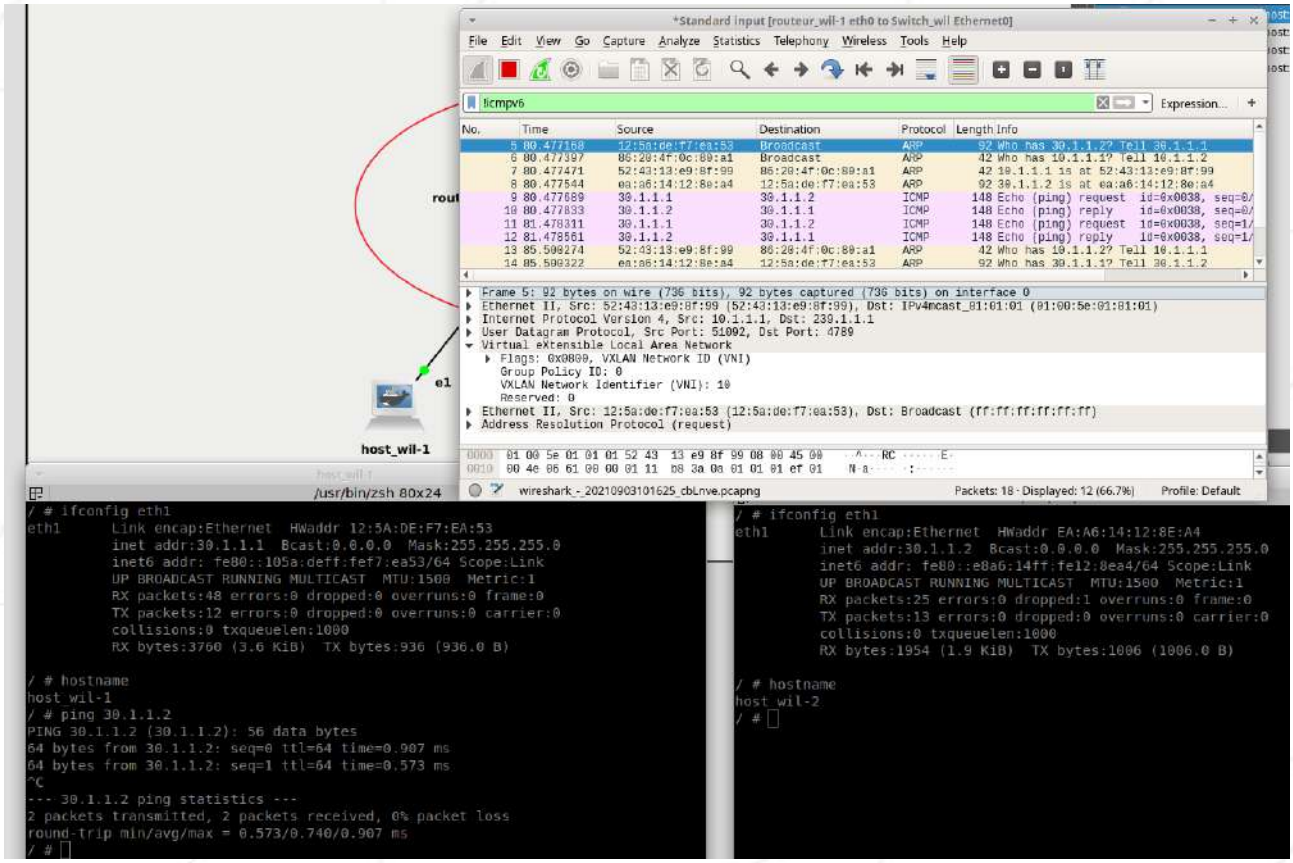
You now have a functional basis to start setting up your first `VXLAN` (RFC 7348) network. First in static then in dynamic multicast. Here is the topology of your first `VXLAN`:



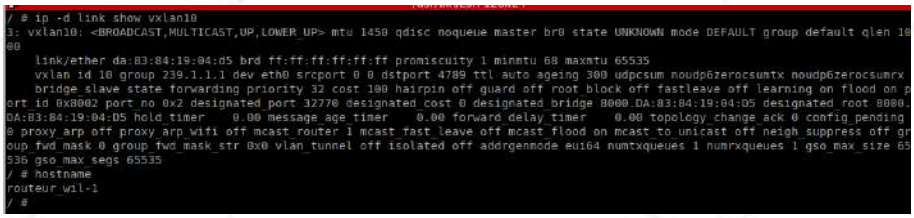
You must configure this network using a `VXLAN` with an ID of `10` as shown in the examples below. You can use any `VXLAN` name you like here: `vxlan10`. You must set up a `bridge` here: br0. You must configure your `ETHERNET` interfaces as you wish. Below is an example of the expected result when we inspect the traffic between our two machines in our `VXLAN`

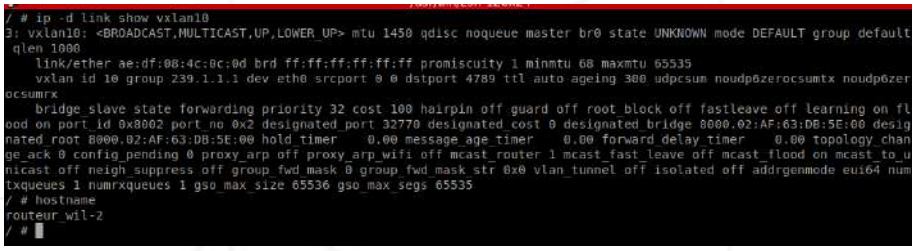


We are now going to see the same thing using the groups whose goal will be to be able to make a dynamic `multicast`.

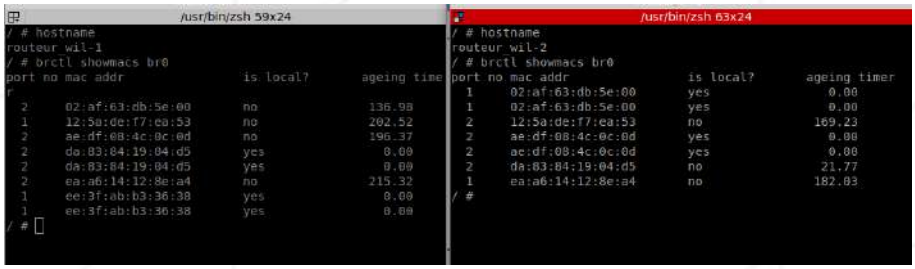


We can notice that our machines now have a group (here 239.1.1.1 you can modify this part):





Below is an example of how to display our mac address table in our two routers:



You must render this project in a `P2` folder at the root of your git repository. You should also add the configuration files with comments to explain the set up of each equipment.

You must export this project with a ZIP compression including the base images. This file must be visible in your git repository.

You must use correct and consistent names for your equipment here with the login of one of the group members.

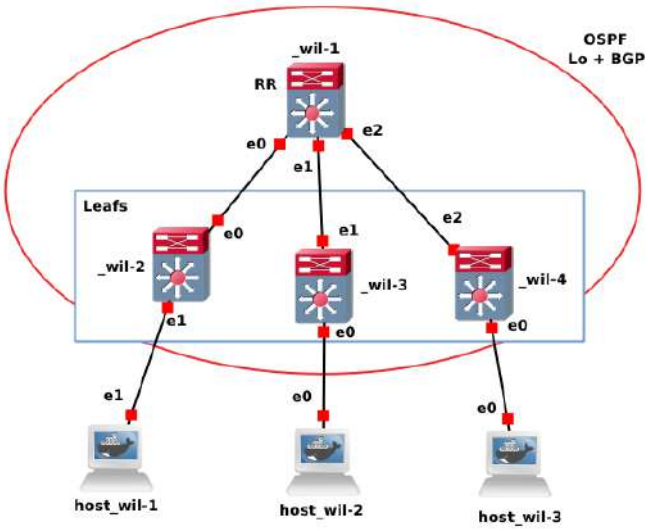
IV.3 Part 3: Discovering BGP with EVPN

Now that you have mastered the basic principle of the `VXLAN` we will go a little further and explore the principle of the `BGP EVPN` (rfc 7432) without using `MPLS` to simplify things. The controller will learn the `MAC addresses`. We will use our `VXLAN` with ID 10 seen in the previous part.

As in the second part we start with the topology of the expected network. We are going to use the principle of the `route reflection` (=RR). Our `leafs` (`VTEP`) will be configured to have `dynamic` relations.

This diagram represents a small datacenter.

`For the sake of readability the names are shorter here. You will have to use OSPF to simplify the evaluation.`

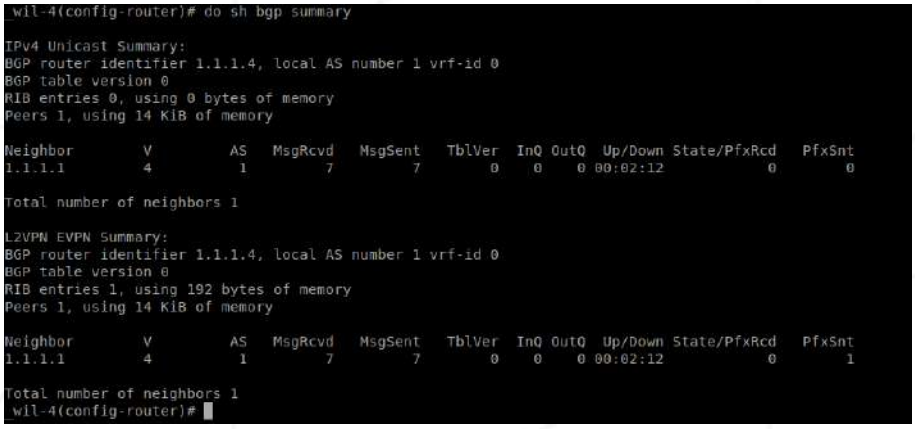


We can see our visibility from our `VTEP _wil-4` the 3 `VTEPs` 1.1.[1.4]

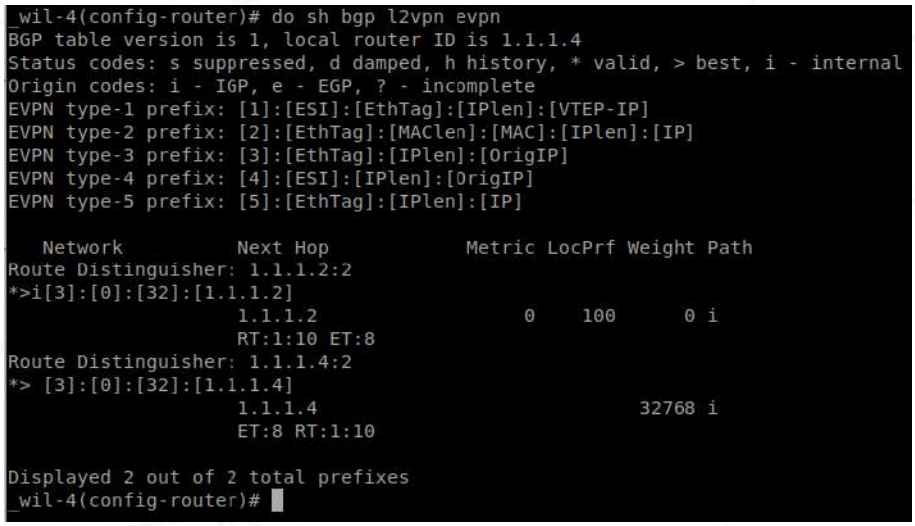
```
_wil-4(config-router)# do sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

O>* 1.1.1.1/32 [110/10000] via 10.1.1.9, eth2, weight 1, 00:00:44
O>* 1.1.1.3/32 [110/20000] via 10.1.1.9, eth2, weight 1, 00:00:44
O   1.1.1.4/32 [110/0] is directly connected, lo, weight 1, 00:00:58
C>* 1.1.1.4/32 is directly connected, lo, 00:00:58
O>* 10.1.1.0/30 [110/20000] via 10.1.1.9, eth2, weight 1, 00:00:44
O>* 10.1.1.4/30 [110/20000] via 10.1.1.9, eth2, weight 1, 00:00:44
O   10.1.1.8/30 [110/10000] is directly connected, eth2, weight 1, 00:00:49
C>* 10.1.1.8/30 is directly connected, eth2, 00:00:58
_wil-4(config-router)#
```

We have only one route for the moment with our controller (RR) :

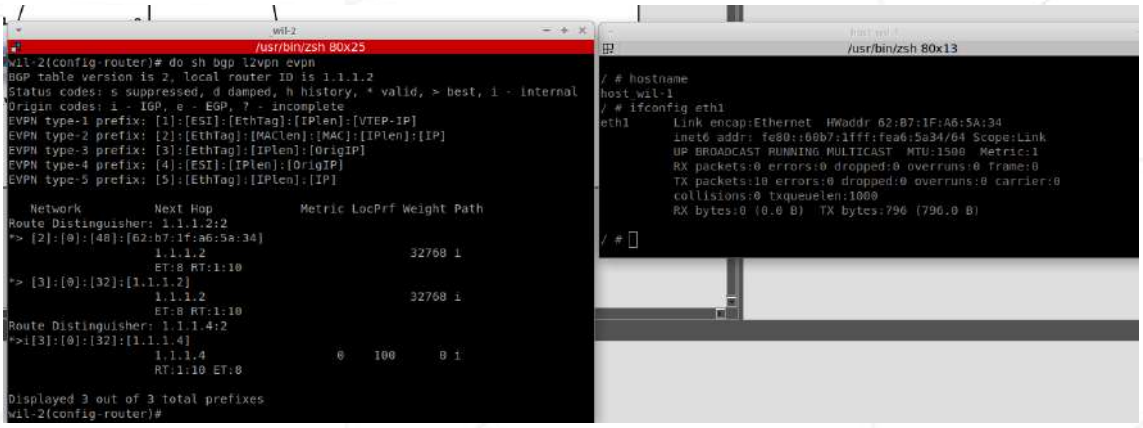


When there is no host running we can see our `VNI` (10 here) as well as our pre-configured routes (`type 3`). No route `type 2` seems to exist and it is quite normal.



A machine `host_wil-1` is now functional. We can notice that without assigning an `IP address` our `VTEP` (`wil_2`) automatically discovers the `MAC address` of the functional machines.

We can also see the automatic creation of a route `type 2`:



In the same way, when we look at a second `VTEP`(`_wil-4`), we can notice the creation of a new route `type 2` generated by our `RR`:

```
_wil-4(config-router)# do sh bgp l2vpn evpn
BGP table version is 2, local router ID is 1.1.1.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[ESI]:[EthTag]:[IPlen]:[VTEP-IP]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.2:2
*>i[2]:[0]:[48]:[62:b7:1f:a6:5a:34]
                    1.1.1.2                  0    100      0 i
                    RT:1:10 ET:8
*>i[3]:[0]:[32]:[1.1.1.2]
                    1.1.1.2                  0    100      0 i
                    RT:1:10 ET:8
Route Distinguisher: 1.1.1.4:2
*> [3]:[0]:[32]:[1.1.1.4]
                    1.1.1.4                            32768 i
                    ET:8 RT:1:10

Displayed 3 out of 3 total prefixes
_wil-4(config-router)#
```

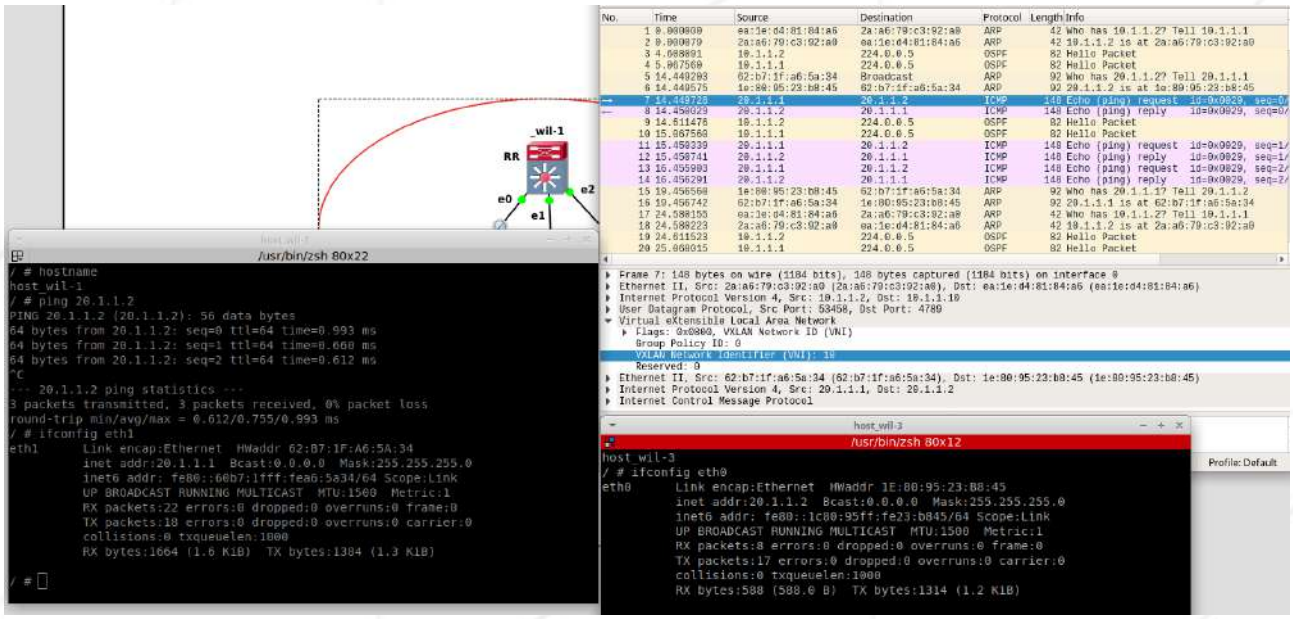
We repeat the operation with a second machine (`host_wil-3`). We can notice the second route set up by `type 2`. There is no assignment of `IP address`:

```
wil-2(config-router)# do sh bgp l2vpn evpn
BGP table version is 2, local router ID is 1.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[ESI]:[EthTag]:[IPlen]:[VTEP-IP]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]

   Network          Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 1.1.1.2:2
*> [2]:[0]:[48]:[62:b7:1f:a6:5a:34]
                    1.1.1.2                            32768 i
                    ET:8 RT:1:10
*> [3]:[0]:[32]:[1.1.1.2]
                    1.1.1.2                            32768 i
                    ET:8 RT:1:10
Route Distinguisher: 1.1.1.4:2
*>i[2]:[0]:[48]:[1e:80:95:23:b8:45]
                    1.1.1.4                  0    100      0 i
                    RT:1:10 ET:8
*>i[3]:[0]:[32]:[1.1.1.4]
                    1.1.1.4                  0    100      0 i
                    RT:1:10 ET:8

Displayed 4 out of 4 total prefixes
wil-2(config-router)#
```

For our verification a simple ping allows us to see that we can access all the machines through our `RR` using the `VTEPs`. We can see the `VXLAN` configured to 10 as well as our packets `ICMP`. We also see packets `OSPF` configured:



You must render this project in a `P3` folder at the root of your git repository. You should also add the configuration files with comments to explain the set up of each equipment.

You must export this project with a ZIP compression including the image bases. This file must be visible in your git repository.

You must use correct and consistent names for your equipment here with the login of one of the group members.

Chapter V

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your folders and files to ensure they are correct.

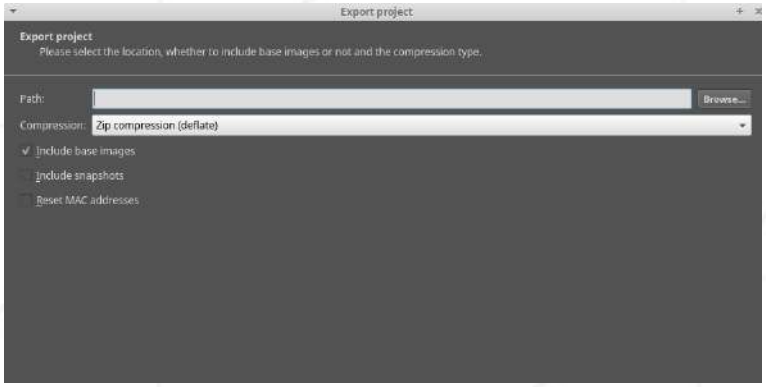
Reminder:

- Turn the mandatory part in three folders located at the root of your repository: `P1`, `P2` et `P3`.

Below is an example of the expected directory structure:

```
> find -maxdepth 2 -ls
  424242     4 drwxr-xr-x  6 wandre  wil42      4096 sept. 17 23:42 .
  424242     4 drwxr-xr-x  3 wandre  wil42      4096 sept. 17 23:42 ./P1
  424242     4 -rw-r--r--  1 wandre  wil42      XXXX sept. 17 23:42 ./P1/P1.gns3project
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P1/_wil-1_host
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P1/_wil-2
  424242     4 drwxr-xr-x  3 wandre  wil42      4096 sept. 17 23:42 ./P2
  424242     4 -rw-r--r--  1 wandre  wil42      XXXX sept. 17 23:42 ./P2/P2.gns3project
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P2/wil-1_g
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P2/_wil-1_host
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P2/_wil-1_s
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P2/_wil-2_g
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P2/_wil-2_host
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P2/_wil-2_s
  424242     4 drwxr-xr-x  3 wandre  wil42      4096 sept. 17 23:42 ./P3
  424242     4 -rw-r--r--  2 wandre  wil42      4096 sept. 17 23:42 ./P3/P3.gns3project
  424242     4 -rw-r--r--  2 wandre  wil42      4096 sept. 17 23:42 ./P3/_wil-1
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P3/_wil-1_host
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P3/_wil-2
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P3/_wil-2_host
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P3/_wil-3
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P3/_wil-3_host
  424242     4 -rw-r--r--  2 wandre  wil42      XXXX sept. 17 23:42 ./P3/_wil-4
> file P3/P3.gns3project
  P3/P3.gns3project: Zip archive data, at least v2.0 to extract
```

To export your projects in zip format go to the menu file then export portable project:



During the evaluation you will have to explain the terms used in the subject. We strongly encourage you to take time to understand each of these.

The evaluation process will happen on the computer of the evaluated group.