

# Documentation - Projet NAS

Adele Prouvost - Titouan-Joseph Revol - Etienne Saisset - Clement Lagneau-Donzelle

Janvier 2021

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Utilisation</b>	<b>3</b>
2.1	Initialisation des configurations . . . . .	3
2.2	Compléter une configuration manuellement . . . . .	3
2.3	Test automatisés . . . . .	3
2.4	Réinitialisation . . . . .	3
<b>3</b>	<b>Structure du fichier de configuration</b>	<b>4</b>
<b>4</b>	<b>Technologies, protocoles et nomenclature mis en place</b>	<b>5</b>
4.1	BGP : Border Gateway Protocol . . . . .	5
4.2	OSPF : Open Shortest Path First . . . . .	5
4.3	MPLS : MultiProtocol Label Switching . . . . .	5
4.4	RP : Routeur Provider . . . . .	5
4.5	PE : Provider Edge . . . . .	5
4.6	CE : Customer Edge . . . . .	5
4.7	VRF : Virtual Routing and Forwarding . . . . .	5

# 1 Introduction

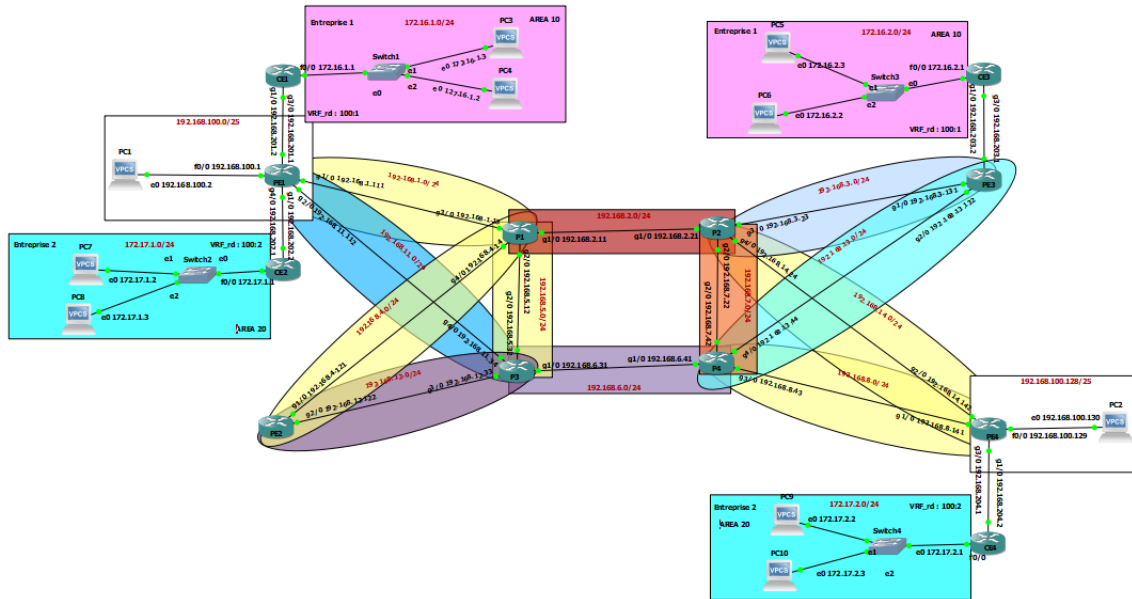


FIGURE 1 – Présentation globale du projet - Configuration "Papillon"

Ce projet a pour but de se familiariser avec les différents protocoles réseau. En Figure 1 une présentation global de notre projet. Le but du projet est d'automatiser une mise en place d'une configuration sur un réseau. Nous utilisons une base de donnée en format JSON qui répertorie les différents protocoles sur chaque routeur ainsi que leur adresses sur chacune des interfaces.

## 2 Utilisation

Nous avons donc développé un script en python3 qui se connecte en telnet sur les routeurs, ce qui permet de mettre les configurations sur chaque routeur.

### 2.1 Initialisation des configurations

1. Récupérer le script sur le github : <https://github.com/titouan-joseph/AutoConfCisco>.
2. Vérifier les ports dans le fichier 'conf.json' ainsi que dans le fichier 'configPC.json'.
3. Vérifier ensuite que tous les routeurs et PC sont allumés.
4. Lancer dans un terminal 'python3 main\_argv.py conf.json' pour configurer les routeurs
5. Lancer dans un terminal 'python3 configPC.py' pour configurer les PC

### 2.2 Compléter une configuration manuellement

Si vous souhaitez ajouter une configuration après avoir mis en place vos configuration, suivez les instructions ci-dessous.

- Pour mettre à jour uniquement un routeur : `python3 main_argv.py conf.json -r ROUTER`
- Pour mettre à jour uniquement une interface : `python3 main_argv.py conf.json -i INTERFACENAME`
- Pour mettre à jour uniquement OSPF : `python3 main_argv.py conf.json -o`
- Pour mettre à jour uniquement MPLS : `python3 main_argv.py conf.json -m`
- Pour faire un write sur tous les routeurs : `python3 main_argv.py conf.json -w`
- Pour enlever les configurations : `python3 main_argv.py conf.json -e` (Attention, les VRF restent actives)

### 2.3 Test automatisés

Vous pouvez tester que l'ensemble des configuration est correct en lançant 'python3 config\_ping.py'.

### 2.4 Réinitialisation

A l'heure actuelle, nous testons le projet sur GNS3 et il n'est pas possible de 'reload' les configurations. Nous devons donc éteindre puis rallumer les routeurs et se référer au paragraphe "Initialisation".

En production, il est possible de reload les routeurs, même si cela a évidemment un impact sur le trafic courant (typiquement ne pas faire ça sur tout le coeur de réseau en même temps).

### 3 Structure du fichier de configuration

Le fichier de configuration est construit comme suit :

- Routeur
  - Routeur IP
  - Port
  - Interface
    - interfaceName
    - toRouter
    - IPv4
    - IPv6
    - MPLS
    - OSPF\_area
    - VRF
  - OSPF\_id
  - OSPF\_neighbor
  - ipcef (bool)
  - VRF
    - name
    - rd
    - rt\_import
    - rt\_export
    - OSPF\_process
    - OSPF\_area
    - network
- BGP
  - AS
  - Neighbor
    - AS
    - Addr

## **4 Technologies, protocoles et nomenclature mis en place**

### **4.1 BGP : Border Gateway Protocol**

Ce protocol sert à effectuer un routage entre AS.

### **4.2 OSPF : Open Shortest Path First**

Ce protocol sert à router à l'intérieur d'un AS

### **4.3 MPLS : MultiProtocol Label Switching**

Ce protocol sert à labéliser des paquets pour faire différents routages.  
Cas typique d'usage : VPN et voix sur IP VoIP.

### **4.4 RP : Routeur Provider**

Ce sont sur la Figure 1 les routeurs P1, P2, P3 et P4.  
Ils correspondent aux routeur dans le coeur du réseau.

### **4.5 PE : Provider Edge**

Ce sont sur la Figure 1 les routeurs PE1, PE2, PE3 et PE4.  
Ils correspondent aux routeur dans le coeur du réseau qui sont en périphérie, ils sont connectés aux CE.

### **4.6 CE : Customer Edge**

Ce sont sur la Figure 1 les routeurs CE1, CE2, CE3 et CE4.  
Ils corresponddnt aux routeurs présents dans le réseaux du client.

### **4.7 VRF : Virtual Routing and Forwarding**

En Route-Distinguisher (RD) tous les préfixes sont dans la table BGP et en Route-Target (RT)