Projet d'Interconnexion dans Internet

Projet réalisé dans le cadre de l'UE
Internet et Graphes
2e année – ENSEEIHT

Auteurs :

Rémi Ariau
Malcolm Bertaina
Mano Dinnat Creusier
Hassan El Sahily
Marina Galstyan
Léo Vanhaecke

Encadrants :

Chaput Emmanuel
Ojo Juliana
Verdier Antonin

ENSEEIHT – Toulouse INP
Département Sciences du Numérique
Année universitaire 2025–2026

Table des matières

1 Introduction

Dans le cadre de l'unité d'enseignement *Internet et Graphes*, ce projet a pour objectif de mettre en pratique les notions fondamentales liées aux réseaux IP, au routage et aux services réseau. Il s'inscrit dans une démarche proche de situations réelles rencontrées dans l'ingénierie des réseaux, en particulier la conception et le déploiement d'un réseau d'entreprise interconnecté à d'autres systèmes autonomes, dans une architecture inspirée du fonctionnement d'Internet.

Le travail demandé consiste à concevoir et mettre en œuvre un système autonome (AS) jouant le rôle de fournisseur de services réseau. Cet AS doit fournir un accès réseau à des clients particuliers et à des entreprises, tout en intégrant des mécanismes essentiels tels que l'adressage IP, le routage dynamique, la sécurité, ainsi que différents services applicatifs. En parallèle, chaque groupe doit déployer un réseau d'entreprise réparti sur deux sites, reliés par un lien sécurisé, et permettre l'accès distant de clients particuliers à ce réseau.

Au-delà de l'aspect purement fonctionnel, l'objectif principal du projet est de comprendre et d'expliquer les choix techniques réalisés. Les décisions concernant l'architecture, les protocoles utilisés ou encore la configuration des services doivent être justifiées et analysées, même lorsque certaines fonctionnalités restent partielles ou en cours de mise en œuvre. Le projet met ainsi l'accent sur la cohérence globale du réseau, la clarté de la conception et la capacité à observer et interpréter le comportement des protocoles déployés.

Ce rapport présente l'architecture retenue, la mise en œuvre du système autonome, l'organisation des réseaux d'entreprise ainsi que les services et mécanismes de sécurité associés. Les choix techniques réalisés sont détaillés et discutés, et les limites rencontrées au cours du projet sont également abordées. Enfin, ce document vise à fournir une vision claire et structurée du travail réalisé.

2 Présentation globale de l'architecture

2.1 Vue d'ensemble

Afin d'établir l'architecture globale de notre réseau, nous nous sommes basés sur les éléments du sujet. Nous avons donc :

- Notre **AS** est composé de quatre routeurs afin de pouvoir mettre en place `OSPF` et simuler un routeur de coeur, un routeur de bordure vers le site d'entreprise principal, un routeur de bordure vers l'AS suivant ainsi qu'un routeur relié aux particuliers et clients. Ils permettent la communication, notamment via VPN entre les différents clients et les entreprises. L'AS accueille également deux serveurs. Tout d'abord, le serveur DNS principal qui permet aux différents terminaux de l'entreprise ainsi qu'aux particuliers de résoudre les noms de domaine de l'AS. Le second serveur n'a pas été utilisé.
- Un **site principal** sur lequel résident la majorité des services. On y trouve pour simplifier, un seul routeur, connecté à un switch lui-même relié à plusieurs postes. Un serveur DNS / VPN permet de résoudre les noms de domaine locaux (et au besoin transmettre les requêtes au serveur principal de l'AS) et autorise la communication avec les employés à distance et les clients particuliers de l'AS. Un autre serveur met en place un service Web, une gestion des utilisateurs avec LDAP, et un service applicatif VoIP qui offre la téléphonie aux employés.
- Un **employé remote** de l'entreprise. Il peut accéder au site principal grâce au VPN et obtient son adresse dynamiquement grâce à sa box.
- Un **client particulier** de l'AS qui accède à l'AS de manière plus restrictive, mais toujours en utilisant une box. Il peut par exemple utiliser le service DNS principal.

— Un **site secondaire**. Comme expliqué plus loin, il n'a pas été totalement configuré mais permet en théorie d'offrir des services similaires au site principal, accessible grâce à un VPN pour les employés d'un autre AS.

2.2 Topologie globale

Vous trouverez ci-dessous le schéma de notre projet (**Désolé, il faut zoomer**).

La partie "vide" en rouge représente l'AS d'un autre groupe, avec lequel nous pouvons communiquer via notre routeur `AS-R-ASx` et accéder à leur site secondaire (que nous accueillons normalement avec leurs adresses et configurations) grâce au service VPN.

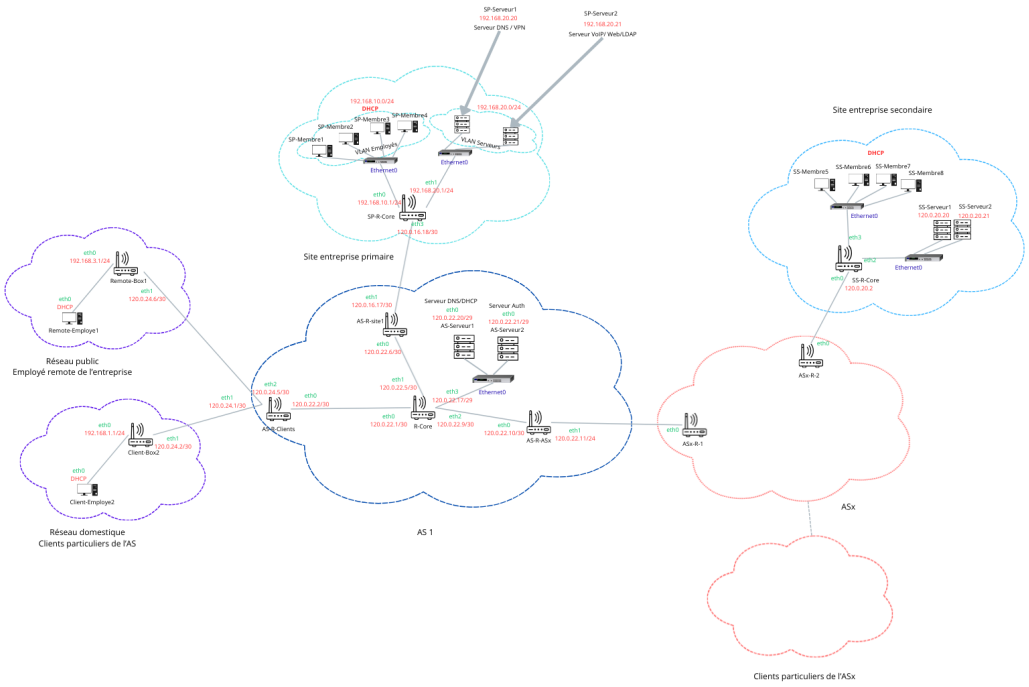


FIGURE 1 – Schéma d'interconnexion de l'AS

2.3 Choix technologiques et outils

Pour ce projet, nous avons décidé d'utiliser une combinaison de GNS3 et Docker. GNS3 permet de visualiser la topologie et de gérer intuitivement le démarrage et l'arrêt de certains composants. Il offre en clair un aspect *Lab* et "Travaux Pratiques". Docker nous permet d'utiliser des conteneurs légers et d'interagir avec notre machine afin de sauvegarder notre travail.

2.3.1 Conteneurs utilisés

Les conteneurs utilisés sont `Alpine` pour les serveurs et les box, ainsi que `FRRouting` pour les routeurs. En effet, les images Cisco ne sont pas libres et sont lourdes en RAM. Pour les services : le DHCP utilise l'outil `dnsmasq`, VoIP utilise `asterisk`, le VPN utilise `wireguard` et le Web `apache`.

2.3.2 Scripts de démarrage et de configuration

Pour sauvegarder nos configurations de manière persistante, nous avons développé des scripts bash utilisant les commandes principales de Docker comme `docker exec` ou `docker cp`. La pipeline est la suivante, après avoir

configuré des équipements sur GNS3 :

- `sync_to_git.sh` permet de sauvegarder les dossiers et fichiers utiles comme `/etc`, `config.sh`, `frr.conf` ou `dnsmasq.conf` dans des dossiers associés à chaque élément (conteneur) du réseau. Ensuite, le travail peut être `push` sur GitHub.
- `sync_from_git.sh` permet d'injecter ces dossiers et fichiers au bon endroit dans les conteneurs Docker de GNS3.
- `execute_config.sh` permet d'exécuter les fichiers `config.sh` de chaque équipement et donc de lancer les démons `frr`, de lancer `asterisk`, de donner les adresses via DHCP, etc.

2.3.3 Docker compose

Le développement de ces scripts s'est avéré très long et complexe (s'intéresser aux scripts Python des démons de frr...). En réalité, nous aurions pu simplement utiliser `docker compose`. Malheureusement, nous ne connaissions pas encore cette fonctionnalité. Cependant, nous avons vu sur Internet que cela intéressait beaucoup de personnes de pouvoir mettre en place des sauvegardes persistantes sur GNS3, qui est beaucoup utilisé pour réaliser des travaux pratiques. La solution que nous proposons est robuste car elle permet de tout réinitialiser à chaque démarrage, ne laissant aucune configuration résiduelle et offrant des logs plus parlants pour déboguer. Maintenant que ces scripts sont disponibles, ils pourraient même être réutilisés facilement pour d'autres projets.

3 Mise en œuvre du Système Autonome (AS)

3.1 Rôle et missions de l'AS

Comme mentionné plus haut, l'AS est un fournisseur de services : accès Internet, transport inter-sites, services (DNS). Il est donc au coeur de notre réseau, car presque toutes les données transitent par ses routeurs et serveurs. L'AS est en effet directement relié aux particuliers, aux employés distants de l'entreprise ainsi qu'aux sites principaux et secondaires. Pour interconnecter ces différents équipements, nous avons réalisé un plan d'adressage qui détaille, pour chaque composante du réseau, la plage d'adresses IP qui lui est attribuée.

Il est **très important** de préciser que certaines plages sont petites (comme les liens point-à-point pour le VPN), tandis que d'autres comme celle du réseau d'entreprises sont plus grandes. Toutes les adresses ne sont bien évidemment pas utilisées mais notre objectif dans ce projet était aussi de volontairement "simuler" un **passage à l'échelle**, comme pour un vrai réseau. Autrement, nous aurions pu tout configurer de manière statique...

3.2 Plan d'adressage IP

Voici le plan d'adressage IP de la plage `120.0.16.0/20 - 120.0.32.0/20`

Table 1 – Plan d'adressage réseau du projet

Réseau	Masque	Première Adr.	Dernière Adr.	Broadcast	Usage	Type Config.
120.0.16.0	/22	120.0.16.1	120.0.19.254	120.0.19.255	Site principal	Statique + DHCP
120.0.20.0	/23	120.0.20.1	120.0.21.254	120.0.21.255	Site secondaire	Statique + DHCP
120.0.22.0	/24	120.0.22.1	120.0.22.254	120.0.22.255	AS	Statique
120.0.24.0	/23	120.0.24.1	120.0.25.254	120.0.25.255	Particuliers	DHCP
120.0.28.0	/26	120.0.28.1	120.0.28.62	120.0.28.63	VPN Client-Site	Statique
120.0.29.0	/28	120.0.29.1	120.0.29.14	120.0.29.15	VPN Site-Site	Statique
120.0.31.0	/30	120.0.31.1	120.0.31.2	120.0.31.3	Lien inter-AS	Statique

3.3 Routage interne à l'AS

Pour le routage, nous utilisons comme demandé un protocole dynamique : `OSPF`. Même si ce protocole est plus lourd que `RIP` en ressources, ce choix est justifié ici 3.

Les zones sont réparties comme suit (rappel de l'architecture 1) :

— **Zone 0 (backbone)** : AS
— **Zone 1** : site de l'entreprise principale
— **Zone 2** : réseau domestique d'un particulier de l'AS
— **Zone 3** : réseau de l'employé *remote* de l'entreprise

Encore une fois, ces zones sont justifiées par le fait qu'elles peuvent être facilement étendues pour accueillir davantage d'équipements.

Voici les routes obtenues, par exemple sur le routeur `SP-R-Core` qui en possède le plus (avec le réseau privé et l'accès à l'AS)

```
/ # ip route
default via 192.168.122.1 dev eth5
120.0.16.16/30 dev eth3 scope link  src 120.0.16.18
120.0.22.0/30 via 120.0.16.17 dev eth3  metric 20
120.0.22.0/24 via 120.0.16.17 dev eth3  metric 20
120.0.22.4/30 via 120.0.16.17 dev eth3  metric 20
120.0.22.8/30 via 120.0.16.17 dev eth3  metric 20
120.0.22.16/29 via 120.0.16.17 dev eth3  metric 20
120.0.24.0/30 via 120.0.16.17 dev eth3  metric 20
120.0.24.4/30 via 120.0.16.17 dev eth3  metric 20
192.168.0.0/24 via 192.168.122.1 dev eth5
192.168.10.0/24 dev eth0 scope link  src 192.168.10.1
192.168.20.0/24 dev eth1 scope link  src 192.168.20.1
192.168.122.0/24 dev eth5 scope link  src 192.168.122.100
```

3.4 Accès des clients particuliers

Afin de fournir un accès internet aux clients particuliers, nous avons mis en place des routeurs type "box internet". Ces routeurs sont basés sur une image Alpine.

La configuration de ces équipements repose sur l'implémentation de deux services :

— **DHCP :** L'outil `dnsmasq` a été configuré pour assurer l'attribution dynamique des adresses IP. Il permet de délivrer des adresses IP privées aux machines connectées au réseau local du client.
— **NAT :** L'outil `iptables` est utilisé pour configurer le NAT. Une règle *MASQUERADE* a été appliquée sur l'interface de sortie afin de permettre aux équipements du réseau privé de communiquer avec le réseau de l'AS en utilisant l'adresse IP publique du routeur.

Le **DHCP** fonctionne parfaitement. Nous pourrions par exemple rajouter un lien entre une nouvelle box reliée à une machine et le routeur `AS-R-Clients`. Le routeur apprendrait la route vers la box internet grâce à OSPF, la box obtiendrait une adresse publique distribuée par le serveur de l'AS (`AS-R-Clients` ayant été configuré en tant que `dhcp-helper`) et enfin la box donnerait une adresse privée à l'utilisateur.

Exemple : la commande `ip a` sur `Client-Box2` montre l'adresse publique obtenue du serveur :

```
108: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel qlen 1000
```

```
    link/ether 02:42:0e:60:82:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 scope global eth0
       valid_lft forever preferred_lft forever
111: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel qlen 1000
    link/ether 02:42:0e:60:82:01 brd ff:ff:ff:ff:ff:ff
    inet 120.0.24.2/30 brd 120.0.24.3 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::42:eff:fe60:8201/64 scope link
       valid_lft forever preferred_lft forever
```

De même, `ip a` sur `Client-Employe2` montre l'adresse privée obtenue de la box :

```
107: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel qlen 1000
link/ether 02:42:81:93:44:00 brd ff:ff:ff:ff:ff:ff
inet 192.168.1.28/24 brd 192.168.1.255 scope global eth0
   valid_lft forever preferred_lft forever
```

Enfin, voici la configuration DNS de `AS-Serveur1`

```
# Lan clients de l'entreprise

dhcp-range=set:clients,120.0.24.2,120.0.24.2,255.255.255.252,24h
dhcp-option=tag:clients,option:dns-server,120.0.22.20
dhcp-option=tag:clients,option:router,120.0.24.1

# Lan employes de l'entreprise

dhcp-range=set:employes,120.0.24.6,120.0.24.6,255.255.255.252,24h
dhcp-option=tag:employes,option:dns-server,120.0.22.20
dhcp-option=tag:employes,option:router,120.0.24.5
```

3.5 Accès des clients entreprises et Qualité de Service (QoS)

Les clients particuliers et les entreprises n'ont pas les mêmes exigences en matière de qualité de service. Les particuliers utilisent un accès de type *best-effort*, sans garantie de débit ni de latence, tandis que les entreprises hébergent des services critiques tels que des serveurs applicatifs, des accès VPN ou des services de téléphonie (VoIP), nécessitant des performances plus stables.

Il est donc nécessaire de différencier le traitement des flux afin de garantir un niveau de service adapté aux besoins de chaque type de client.

3.5.1 Séparation des flux

Dans notre architecture, le routeur `R-Core` est le seul point où les flux des clients particuliers et des entreprises se mélangent avant d'emprunter les mêmes liens de sortie vers les sites d'entreprise. Les autres routeurs du système autonome ne transportent qu'un type de trafic spécifique et ne constituent donc pas des points de congestion pertinents pour l'application de la QoS.

La séparation logique des flux est ainsi réalisée au niveau du routeur cœur, qui représente le point de concurrence réel entre les différents types de trafic.

3.5.2 Mise en place de la QoS

La QoS est mise en œuvre sur les interfaces de sortie du routeur `R-Core` à l'aide du mécanisme `HTB` (Hierarchical Token Bucket) du noyau Linux. Ce mécanisme permet de définir plusieurs classes de trafic avec des priorités et des débits garantis.

Trois classes ont été définies :

— une classe à haute priorité pour le trafic critique de l'entreprise (VoIP, VPN, serveurs),

— une classe intermédiaire pour le trafic applicatif standard de l'entreprise,

— une classe à basse priorité pour le trafic des clients particuliers, traité en *best-effort*.

Les paquets sont classifiés à l'aide de filtres basés sur les adresses IP des réseaux et sur les ports des services, puis redirigés vers la classe correspondante.

3.5.3 Tests et validation

Afin de valider la mise en œuvre de la QoS, des tests de trafic concurrent ont été réalisés. Lorsque des flux intensifs sont générés par des clients particuliers, le trafic à destination des sites d'entreprise conserve une bande passante stable et une latence maîtrisée.

Ces tests montrent que les mécanismes de QoS permettent de protéger efficacement le trafic critique de l'entreprise face aux usages intensifs des clients particuliers, conformément aux objectifs du cahier des charges.

4 Interconnexion des Systèmes Autonomes

4.1 Objectifs

L'objectif de l'interconnexion des systèmes autonomes est de permettre la communication entre plusieurs AS indépendants, afin de simuler un Internet simplifié. Cette interconnexion doit permettre :

— l'échange de trafic entre les réseaux d'entreprise de différents groupes,

— l'accès aux services distants (sites secondaires, VPN inter-sites),

— la mise en évidence des problématiques liées au routage inter-domaine.

Dans le cadre de ce projet, l'interconnexion vise principalement à illustrer les mécanismes généraux d'échange de routes entre AS, sans chercher à reproduire l'ensemble de la complexité du routage Internet réel.

4.2 Choix de conception

Compte tenu de la taille réduite de la topologie et du nombre limité de systèmes autonomes impliqués, il a été décidé de ne pas mettre en œuvre le protocole BGP. En effet, dans un contexte aussi simple, l'utilisation de BGP aurait introduit une complexité supplémentaire sans réel bénéfice fonctionnel.

Le choix s'est donc porté sur :

— une interconnexion directe entre AS via des liens point-à-point,

— l'utilisation de routes statiques pour permettre la sortie et l'entrée du trafic inter-AS,

— une topologie partiellement maillée, chaque AS étant relié uniquement à ses voisins immédiats.

Ce choix est conforme aux consignes du projet, qui autorisent des solutions simplifiées lorsque le contexte ne justifie pas l'emploi de protocoles inter-domaine complets.

4.3 Mise en œuvre technique (conception prévue)

L'interconnexion inter-AS devait reposer sur un lien point-à-point entre le routeur de bordure de notre AS (`AS-R-ASx`) et le routeur de bordure d'un AS voisin, conformément au schéma d'interconnexion du projet. L'objectif

était de créer un réseau de transit dédié reliant directement les deux systèmes autonomes.

Dans un scénario simple impliquant un nombre limité d'AS, la solution prévue consistait à utiliser des routes statiques afin de permettre :

— la sortie du trafic vers les préfixes appartenant aux autres AS,

— l'accès aux sites secondaires hébergés par des groupes voisins.

Lorsque les groupes utilisent tous GNS3, l'interconnexion peut être réalisée en important les projets respectifs et en reliant directement les routeurs de bordure au sein d'une même topologie. Cette approche permet une interconnexion entièrement simulée et un contrôle total du routage inter-AS.

Cependant, tous les groupes n'utilisent pas nécessairement le même environnement de simulation. Une solution alternative, envisagée mais non implémentée, consiste à utiliser la connectivité de la machine hôte via les mécanismes de NAT de GNS3 et de port forwarding. Dans ce cas, le routeur de bordure de l'AS est connecté à l'interface réseau de l'hôte, permettant de faire transiter le trafic inter-AS via le réseau physique (par exemple le Wi-Fi). Cette méthode permettrait d'interconnecter des projets hétérogènes ou des environnements distincts, au prix d'une perte de contrôle fine sur le routage.

Faute de temps, aucune de ces solutions d'interconnexion inter-AS n'a pu être mise en œuvre. Néanmoins, l'architecture de notre AS (présence d'un routeur de bordure dédié, plan d'adressage cohérent et routage interne fonctionnel) a été conçue pour permettre une intégration rapide de l'interconnexion, quelle que soit la solution retenue.

4.4 Limitations et simplifications

L'interconnexion inter-AS constitue la principale limitation de notre déploiement : elle n'a pas pu être implémentée dans le temps imparti. Par conséquent, les tests de connectivité inter-AS (accès à des réseaux externes, communication avec le site secondaire d'un autre groupe) n'ont pas pu être validés expérimentalement.

De plus, nous n'avons pas mis en place BGP. Dans une architecture Internet réaliste, BGP serait le protocole standard pour l'échange de routes entre AS, permettant notamment des politiques de routage (filtrage, préférences, etc.). Cependant, dans le cadre de ce projet et au vu du scénario très simplifié, l'usage de **routes statiques** constituait une approche suffisante et plus rapide à déployer (consigne du prof).

5 Réseau d'entreprise – Site principal

5.1 Présentation

Selon nous, là où l'AS représentait surtout le coeur du réseau (fournir l'accès à Internet) et relayait les informations, le site principal du réseau d'entreprise est un réel **fournisseur de services**. Comme mentionné plus haut, le site principal est dans la **Zone 1** (OSPF). L'unique routeur `SP-R-Core` est relié à la fois à l'AS en communiquant avec `AS-R-site1` 1, mais aussi aux switchs du réseau local et à l'interface `virbr0` de l'ordinateur afin d'avoir un vrai accès à internet pour les tests, notamment de VoIP (via le NAT).

Les services suivants (les applications notamment) sont complètement implémentés dans le site principal : VPN (pour la communication avec les employés), VoIP, DNS (local), DHCP (local), Serveur Web. Ces services sont présentés dans les sections qui suivent.

5.2 Adressage dynamique (DHCP)

Ici, le fonctionnement est légèrement différent de la section évoquant déjà DHCP 3.3. Nous n'avons pas d'accès "box-like". Le routeur possède deux de ses interfaces (`eth0` et `eth1`) dans deux réseaux locaux : `192.168.10.0/24`

pour les employés et `192.168.20.0/24` pour les serveurs 1. Le routeur n'est donc pas un intermédiaire, ces interfaces sont configurées statiquement, et il distribue directement des adresses IP privées aux machines.

5.2.1 Configuration DHCP du routeur

```
interface=eth0
bind-interfaces

no-resolv

# Envoyer toutes les requetes inconnues vers le DNS de l'AS
server=120.0.22.20

# Envoyer le domaine entreprise vers le serveur DNS interne
server=/entreprise.local/192.168.20.20

# Autoriser les requetes venant de l'exterieur (WAN) pour la deleguation
interface=eth3
interface=eth0
interface=eth1

dhcp-range=192.168.10.10,192.168.10.100,255.255.255.0,24h
dhcp-option=option:dns-server,192.168.20.20
dhcp-option=option:router,192.168.10.1
```

5.2.2 Configuration statique du routeur

```
#!/bin/sh

# On clean
killall udhcpc 2>/dev/null
ip addr flush dev eth0
ip addr flush dev eth1
ip addr flush dev eth5
ip route del default 2>/dev/null

ip link set eth5 up
udhcpc -i eth5

apk update
apk add iptables dnsmasq tcpdump

# Retablissement de l'adressage statique
killall udhcpc
ip addr flush dev eth5
ip route del default 2>/dev/null

ip addr add 192.168.122.100/24 dev eth5

# Configuration NAT
ip route add default via 192.168.122.1 dev eth5
ip route add 172.22.222.239/22 via 192.168.122.1 dev eth5


# Configuration LAN Employes
```

```
ip addr add 192.168.10.1/24 dev eth0
ip link set eth0 up

# Configuration LAN Serveurs
ip addr add 192.168.20.1/24 dev eth1
ip link set eth1 up

# Configuration cote WAN - STATIQUE
ip link set eth3 up

# Routage et NAT
echo 1 > /proc/sys/net/ipv4/ip_forward #activation du routage

...
```

5.2.3 Test de DHCP sur le site principal

`SP-Membre1` obtient alors également une adresse :

```
64: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel qlen 1000
    link/ether 02:42:10:e1:c9:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.24/24 brd 192.168.10.255 scope global eth0
       valid_lft forever preferred_lft forever
```

De plus, nous pouvons logiquement ping par exemple le serveur `AS-Serveur1` depuis `SP-Membre1`

```
/etc # ping dns.as.local
PING dns.as.local (120.0.22.20): 56 data bytes
64 bytes from 120.0.22.20: seq=0 ttl=61 time=0.362 ms
64 bytes from 120.0.22.20: seq=1 ttl=61 time=2.901 ms
^C
--- dns.as.local ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.362/1.631/2.901 ms
/etc #
```

5.3 Sécurité interne

La sécurité de l'entreprise repose sur le principe du "moindre privilège". En effet, nous bloquons tout par défaut et n'autorisons que le trafic nécessaire.

1. Isolation des réseaux

Tout d'abord, l'entreprise est divisée en zones (LAN Employés, LAN Serveurs, WAN). On filtre ce qui passe d'une zone à une autre. Par exemple, nous n'autorisons que le Web (`80`), le DNS (`53`) et VoIP (`5060 / 10000-20000`) vers les serveurs.

```
iptables -A FORWARD -i eth0 -o eth1 -d 192.168.20.21 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -p udp --dport 53 -d 192.168.20.20 -j ACCEPT
iptables -A FORWARD -i eth5 -o eth1 -p udp -d 192.168.20.21 --dport 5060 -j ACCEPT
iptables -A FORWARD -i eth5 -o eth1 -p udp -d 192.168.20.21 --dport 10000:20000 -j ACCEPT
...
```

2. Protection contre l'extérieur

Le NAT (en utilisant `MASQUERADE`) est aussi une sécurité : il rend les machines internes invisibles depuis Internet, c'est ce que nous utilisons par exemple pour VoIP.

```
iptables -t nat -A POSTROUTING -o eth3 -j MASQUERADE
...
```

3. Contrôle des flux d'état

On garantit qu'un pirate externe ne peut pas entrer si une machine interne n'a pas initié la connexion :

```
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
```

4. Accès distant sécurisé (VPN)

Pour ne pas ouvrir les services à n'importe qui sur internet, on force l'utilisation d'un tunnel chiffré (VPN)

```
iptables -t nat -A PREROUTING -i eth3 -p udp --dport 51820 -j DNAT --to-destination 192.168.20.20
```

5.4 DNS de l'entreprise

Pour configurer le DNS, nous utilisons les deux serveurs `SP-Serveur1` (de l'entreprise) ainsi que `AS-Serveur1` (de l'AS). Le principe est le suivant : un utilisateur émet une requête DNS. Elle est envoyée à `SP-Serveur1`. Si ce dernier connaît le nom de domaine, il lui renvoie la réponse. Sinon, il relaye la requête à `SP-R-Core` qui la transmettra lui-même à `AS-Serveur1`. On peut également faire des requêtes au serveur de l'AS depuis les clients ou particuliers.

5.4.1 Configuration DNS de `AS-Serveur1`

```
dhcp-authoritative
log-dhcp

interface=eth0
bind-interfaces
domain-needed
bogus-priv

domain=as.local
expand-hosts
local=/as.local/

no-resolv
server=/entreprise.local/120.0.16.18
server=8.8.8.8

cache-size=1000

address=/dns.as.local/120.0.22.20
address=/serveur.as.local/120.0.22.21
address=/routeur-clients.as.local/120.0.22.2
address=/routeur-entreprise.as.local/120.0.22.6
address=/routeur-coeur.as.local/120.0.22.1
address=/routeur-passerelle.as.local/120.0.22.10
```

5.4.2 Configuration DNS de `SP-Serveur1`

```
interface=eth0
bind-interfaces
domain-needed
bogus-priv
no-resolv

domain=entreprise.local
expand-hosts
local=/entreprise.local/
server=/as.local/120.0.22.20


address=/serveur-web.entreprise.local/192.168.20.21
address=/dns.entreprise.local/192.168.20.20
address=/routeur.entreprise.local/192.168.20.1

# On redirige le trafic vers le routeur SP-R-Core qui heberge aussi
# un service dnsmasq (et pourra rediriger vers le serveur de l'AS)
server=192.168.20.1
```

5.4.3 Test d'un serveur DNS

Nous pouvons par exemple correctement résoudre le nom du serveur web.

```
/etc # nslookup serveur-web.entreprise.local
Server:     192.168.20.20
Address:    192.168.20.20:53

Name:   serveur-web.entreprise.local
Address: 192.168.20.21
```

5.5 Service VoIP

La mise en place du service VoIP a été particulièrement difficile. Détailler ou donner toutes les commandes et scripts associés serait trop long. Cependant, nous expliquons ici le fonctionnement global. Finalement, nous avons réussi à le faire marcher sur de vrais téléphones en passant par le WIFI et en utilisant l'application MizuDroid disponible sur le PlayStore (Android).

5.5.1 Configuration

Il y a principalement trois étapes essentielles derrière le fonctionnement de VoIP : la transmission du signal, la traduction par l'ordinateur et le routeur ainsi que le traitement effectué par `asterisk`. On suppose deux utilisateurs ayant pour identifiants `7001` et `7002`. L'ordinateur physique à l'IP `192.168.0.28` sur son interface wifi et le serveur VoIP à l'IP `192.168.20.21`.

1. Transmission du téléphone à l'ordinateur

On configure MizuDroid pour que tous les appels passent par l'ordinateur physique. Quand on passe un appel avec un téléphone physique (`7001`), le téléphone envoie un message (`SIP`) à l'IP du PC (`192.168.0.28`) qui est vu comme le serveur. Il utilise le port `5061` (que l'on a ouvert sur Linux).

2. Redirection (DNAT) dans le réseau privé

Grâce à plusieurs règles `iptables` que nous mettons en place sur l'ordinateur, nous autorisons et redirigeons ces messages vers le routeur en passant par l'interface utilisée par GNS3 (`virbr0`).

```
sudo iptables -t nat -I PREROUTING -i $IF_WIFI -p udp --dport 5061 -j DNAT --to-destination
    $IP_ROUTEUR:5060
sudo iptables -t nat -I PREROUTING -i $IF_WIFI -p udp --dport 10000:20000 -j DNAT --to-destination
     $IP_ROUTEUR
...
```

Puis, le paquet arrive sur l'interface `eth5` du routeur. Ce dernier change l'adresse de destination en la remplaçant par l'IP réelle du serveur grâce à la règle de `DNAT`.

```
iptables -t nat -A PREROUTING -i eth5 -p udp --dport 5060 -j DNAT --to-destination 192.168.20.21
...
```

Enfin, le routeur `forward` ce paquet vers eth1 (le LAN serveur).

3. Traduction d'adresses (NAT et `externaddr`)

Comme l'IP `192.168.20.21` n'est pas routable depuis le réseau WIFI, on a configuré `asterisk` afin qu'il indique dans ses messages qu'il se présente comme étant publiquement `192.168.0.28` lorsqu'il communique. Puis le téléphone reçoit cette information et sait exactement où envoyer le son (flux RTP) et les confirmations (ACK).

```
; ---- Configuration Reseau ----
localnet=192.168.20.0/255.255.255.0
externaddr=192.168.0.28:5061
externtlport=5061
```

5.5.2 Observations

Dans la configuration du serveur, nous ajoutons deux utilisateurs (ils sont maintenant gerés par le LDAP).

```
[7001]
type=friend
host=dynamic
secret=7001
context=internal
qualify=yes
nat=force_rport,comedia
directmedia=no
```

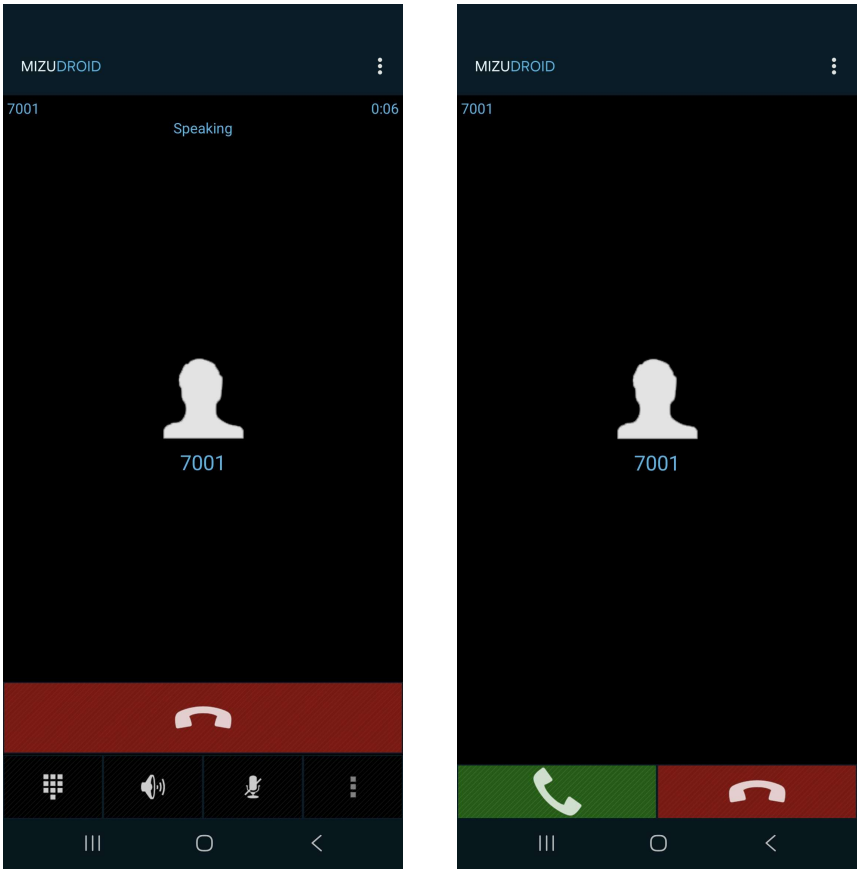
```

[7002]
type=friend
host=dynamic
secret=7002
context=internal
directmedia=no
nat=force_rport,comedia
qualify=yes
```

En tapant `asterisk -r` puis `sip show peers` sur le serveur nous voyons que les deux utilisateurs sont connectés. On remarque que l'IP affichée pour les deux utilisateurs est `192.168.122.1`. C'est tout à fait normal : c'est l'IP de

la passerelle `virbr0` du PC vers GNS3. Puisque l'on fait du NAT sur le PC, `asterisk` voit tous les appels extérieurs comme venant du PC et non directement du téléphone.

```
Connected to Asterisk 20.11.1 currently running on SP-Serveur2 (pid = 397)
SP-Serveur2*CLI> sip show peers
Name/username              Host                                    Dyn Forcerport Comedia    ACL
    Port     Status      Description
7001/7001                  192.168.122.1                            D  Yes        Yes
    4967     OK (1015 ms)
7002/7002                  192.168.122.1                            D  Yes        Yes
    11578    OK (12 ms)
2 sip peers [Monitored: 2 online, 0 offline Unmonitored: 0 online, 0 offline]
```



Appelant (7001) Appelé (7002)

FIGURE 2 – Illustration de VoIP

5.6 Mise en œuvre de l'authentification centralisée (Asterisk Realtime)

L'objectif de cette partie est de séparer la gestion des comptes utilisateurs du service de téléphonie. Plutot que de définir les comptes SIP statiquement dans les fichiers de configuration du serveur (`sip.conf`), nous avons implémenté l'architecture **Asterisk Realtime (ARA)**. Cette architecture permet au serveur Asterisk (SP-Serveur2) d'interroger dynamiquement l'annuaire OpenLDAP pour obtenir les requêtes d'enregistrement (REGISTER) et d'appels (INVITE).

5.6.1 Structure de l'annuaire LDAP

Nous avons préalablement ajouté des comptes utilisateurs dans l'annuaire OpenLDAP, en utilisant le schéma d'objets `asterisk.schema`. Ce schéma étend les classes standard (`inetOrgPerson`) pour inclure les attributs spécifiques à la VoIP.

Voici la structure d'une entrée utilisateur (DN) telle qu'elle est définie dans notre base de données :

```
dn: cn=7001,ou=Employes,dc=entreprise,dc=local
objectClass: top
objectClass: inetOrgPerson
objectClass: asteriskSIPUser
uid: 7001
userPassword: 1234
AstAccountCallerID: "Employe 7001" <7001>
AstAccountContext: internal
AstAccountHost: dynamic
```

5.6.2 Configuration du pilote LDAP dans Asterisk

L'interfaçage repose sur le module `res_config_ldap.so`. La configuration se déroule en deux étapes dans le répertoire `/etc/asterisk/`.

Définition de la connexion et du mapping (`res_ldap.conf`) :

ce fichier établit la liaison réseau avec le service slapd (port 389) et définit la correspondance (mapping) entre les variables internes d'Asterisk et les attributs de l'annuaire.

Configuration appliquée :

```
[general]
url=ldap://127.0.0.1
user=cn=admin,dc=entreprise,dc=local
pass=1234
basedn=dc=entreprise,dc=local

; Mapping des attributs (Asterisk = Attribut_LDAP)
name = uid
sipname = uid
secret = userPassword
context = AstAccountContext
host = AstAccountHost
callerid = AstAccountCallerID
```

5.6.3 Activation du switch Realtime (`extconfig.conf`)

: Ce fichier modifie le comportement du moteur SIP. Nous redirigeons les configuration `sippeers` et `sipusers` vers le driver LDAP configuré ci-dessus, remplaçant ainsi la lecture des fichiers statiques.

```
[settings]
sipusers => ldap,general,sip_users
sippeers => ldap,general,sip_users
```

5.6.4 Tests et Validation

Suite au rechargement des modules (`module reload res_config_ldap.so`), la validation a été effectuée via l'interface en ligne de commande (CLI) d'Asterisk.

Nous avons simulé la récupération des informations pour le pair 7001 :

```
SP-Serveur2*CLI> sip show peer 7001 load

  * Name       : 7001
  Secret       : <Set>
  Context      : internal
  Dynamic      : Yes
  Status       : Unmonitored
```

5.7 Autre service applicatif

Le service applicatif qui a été choisi est un serveur web hébergé sur `SP-Serveur2` (le deuxième serveur du site principal donc). Par défaut, `apache2` recherche les ressources dans `/var/www/localhost/htdocs/`. Un fichier `index.html` a été écrit pour tester.

FIGURE 3 – Page Web du site principal de l'entreprise

Dans la configuration suivante, nous injectons directement le contenu de ce fichier.

```
mkdir -p /run/apache2
mkdir -p /etc/apache2
mkdir -p /var/www/localhost/htdocs

echo "<html><body><h1>Site Officiel de l'Entreprise Groupe 1</h1></body></html>" > /var/www/
    localhost/htdocs/index.html

# Donner un nom de domaine a Apache
echo "ServerName serveur-web.entreprise.local" >> /etc/apache2/httpd.conf

httpd -k start

# Pour etre sur qu'Apache prenne notre page web, et pas une par defaut
chmod -R 755 /var/www/localhost/htdocs
chown -R apache:apache /var/www/localhost/htdocs
```

Nous arrivons correctement à récupérer le contenu à partir d'un hôte !

```
/etc # curl http://serveur-web.entreprise.local
<html><body><h1>Site Officiel de l'Entreprise Groupe 1</h1></body></html>
/etc #
```

6 Réseau d'entreprise – Site secondaire

6.1 Présentation et objectifs

Chaque AS héberge le site secondaire de l'entreprise d'un autre AS. Concrètement, cela signifie que nous accédons à notre site secondaire non pas en utilisant notre plan d'adressage, comme pour le site principal, mais en communiquant avec l'AS voisin et en le "traversant" afin d'accéder à nos services. Nous devons donc échanger nos plans d'adressages avec un autre groupe afin de pouvoir router le trafic destiné à notre site secondaire à travers leur AS. Eux-mêmes routent ensuite ce trafic vers notre site.

6.2 Mise en place du site

Malheureusement, comme discuté dans la conclusion 9.4, nous n'avons pas pu mettre en place l'interconnexion avec d'autres AS. En conséquence, le site secondaire que nous hébergeons ne sert pas réellement. Nous l'avons simplement relié à notre AS. Cependant, les services seraient les mêmes que ceux proposés par notre site principal. Les adresses IP doivent être cependant être adaptées.

7 Accès distant

7.1 VPN employé-entreprise

Un VPN a été mis en place pour permettre l'accès distant au réseau de l'entreprise à un employé.

Pour cela, le protocole `WireGuard` a été privilégié au profit d'autres solutions telles que OpenVPN ou IPsec. WireGuard offre une sécurité moderne et des performances élevés tout en permettant une configuration relativement simple.

7.1.1 Architecture

— **Serveur VPN :** Une machine située dans le LAN des serveurs du site principal (`192.168.20.20`). Elle agit comme point de terminaison du tunnel à l'intérieur de l'infrastructure.
— **Client nomade :** Le poste de l'employé, situé sur le réseau public, initie la connexion.

7.1.2 Configuration technique

Le tunnel VPN utilise un sous-réseau virtuel dédié (`10.99.0.0/24`) pour encapsuler les échanges :
— **Adressage :** Le serveur possède l'adresse virtuelle `10.99.0.1` et le client `10.99.0.2`.
— **NAT sortant :** Une règle de *MASQUERADE* a été appliquée sur le serveur VPN pour permettre aux paquets venant du client distant de naviguer sur le réseau local de l'entreprise comme s'ils venaient du serveur lui-même.

7.1.3 Sécurité

La sécurité de la connexion repose sur l'échange de clés asymétriques. Chaque pair possède sa propre paire de clés privée/publique. L'authentification est stricte : seuls les pairs dont la clé publique est explicitement déclarée dans le fichier de configuration du serveur sont autorisés à établir une connexion.

De plus, des règles de pare-feu (`iptables`) sur le serveur VPN limitent le forwarding, n'autorisant le trafic que vers les services légitimes de l'entreprise.

7.1.4 Tests

1. **Établissement du tunnel :** Vérification du "Handshake" WireGuard via la commande `wg show`.

2. **Accès ICMP :** Test de ping réussi depuis le poste client remote vers les machines du LAN Employés.

```
/ # ping 192.168.10.83
PING 192.168.10.83 (192.168.10.83): 56 data bytes
64 bytes from 192.168.10.83: seq=0 ttl=62 time=5.628 ms
64 bytes from 192.168.10.83: seq=1 ttl=62 time=8.077 ms
64 bytes from 192.168.10.83: seq=2 ttl=62 time=6.213 ms
^C
--- 192.168.10.83 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 5.628/6.639/8.077 ms
/ #
```

7.2 VPN inter-sites

Un tunnel VPN site-à-site a été déployé pour interconnecter le site principal et le site secondaire. Comme pour le VPN employé-entreprise, le protocole `WireGuard` a été retenu.

7.2.1 Architecture

L'interconnexion repose sur deux passerelles VPN agissant comme ponts entre les sites :

— **Serveur Site Principal :** Situé en `192.168.20.20`, il agit comme le nœud central de la topologie VPN.

— **Serveur Site Secondaire :** Situé en `192.168.40.20`, il encapsule tout le trafic du site secondaire à destination du siège.

7.2.2 Configuration technique

Le tunnel utilise le même sous-réseau virtuel dédié (`10.99.0.0/24`) pour le transport :

— **Adressage :** Le serveur principal conserve l'IP `10.99.0.1` et le serveur secondaire prend l'IP `10.99.0.3`.

— **AllowedIPs :** Les fichiers de configuration définissent via le champ `AllowedIPs` les sous-réseaux accessibles de chaque côté (ex : le site B autorise les réseaux `192.168.10.0/24` et `192.168.20.0/24`).

— **Routage :** Des routes statiques ont été ajoutées sur les routeurs de bordure (`SP-R-Core` et `SS-R-Core`) pour rediriger le trafic inter-sites vers leurs serveurs VPN respectifs.

7.2.3 Sécurité

Le niveau de sécurité est identique à celui de l'accès distant, reposant sur des concepts cryptographiques modernes.

7.2.4 Tests

1. **Établissement du tunnel :** Vérification du "Handshake" WireGuard via la commande `wg show`.

```
/ # ip a show eth0
115: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel qlen 1000
    link/ether 02:42:28:bd:74:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.83/24 brd 192.168.10.255 scope global eth0
       valid_lft forever preferred_lft forever
/ # ping 192.168.30.55
PING 192.168.30.55 (192.168.30.55): 56 data bytes
64 bytes from 192.168.30.55: seq=0 ttl=60 time=2.424 ms
64 bytes from 192.168.30.55: seq=1 ttl=60 time=7.998 ms
64 bytes from 192.168.30.55: seq=2 ttl=60 time=8.118 ms
^C
--- 192.168.30.55 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 2.424/6.180/8.118 ms
/ #
```

```
/ # wg show
interface: wg0
  public key: SYVN33ztvGAbSNIgjLlAmzE2P5QOg/JuoNKUB2/rehg=
  private key: (hidden)
  listening port: 51820

peer: OKnlOn/Jk0pf6fXTL1aBplMqi9kj8nBv6G/CLgXQgmY=
  endpoint: 120.0.16.18:51820
  allowed ips: 10.99.0.1/32, 192.168.10.0/24, 192.168.20.0/24
  latest handshake: 1 minute, 35 seconds ago
  transfer: 5.06 KiB received, 3.86 KiB sent
  persistent keepalive: every 25 seconds
/ #
```

2. **Accès ICMP :** Validation de la connectivité ICMP entre une machine du Site A (`192.168.10.x`) et une machine du Site B (`192.168.30.x`).

8 Organisation et Gestion de Projet

8.1 Gestion des sources et organisation du projet

Afin d'assurer la cohérence du travail en groupe, la traçabilité des configurations et la reproductibilité de la plateforme, nous avons adopté une organisation reposant sur deux supports distincts et complémentaires : un dépôt **GitLab de l'INP** pour le projet technique, et un espace **Google Drive** utilisé à des fins organisationnelles internes au groupe.

8.1.1 Dépôt GitLab et organisation du dépôt

Le dépôt GitLab constitue le support central du projet. Il regroupe exclusivement les éléments techniques nécessaires au déploiement et à la reproduction de la plateforme réseau. Il permet de versionner les configurations, de centraliser les scripts et de garantir la cohérence globale de l'architecture.

La structure générale du dépôt est la suivante :

— `GNS3/interco_groupe_1/` : projet GNS3 contenant la topologie du réseau (routeurs, liens, conteneurs).

— `configs/` : configurations persistantes des équipements (routeurs FRRouting, serveurs, services).

— `scripts/` : scripts d'automatisation permettant la sauvegarde, la restauration et l'exécution des configurations dans les conteneurs Docker.

— `README.md` : description générale du projet, prérequis et instructions de lancement.

— `.gitignore` : fichiers et répertoires exclus du versionnement.

Cette organisation permet de séparer clairement la topologie, les configurations réseau et les mécanismes d'automatisation. Les scripts fournis assurent une remise en état rapide de la plateforme après un redémarrage de GNS3, garantissant ainsi la reproductibilité des tests et des démonstrations.

8.1.2 Espace Google Drive

En parallèle du dépôt GitLab, un espace Google Drive partagé a été utilisé de manière **privée** entre les membres du groupe. Cet espace n'est pas destiné à contenir le projet technique en lui-même, mais sert principalement de support d'organisation et de coordination.

Il regroupe notamment :

— des documents de suivi et de répartition du travail,

— des notes personnelles ou collectives,

— des compléments de configuration temporaires,

— des documents intermédiaires facilitant les échanges entre membres.

Le Drive a ainsi permis de fluidifier la communication et l'organisation interne du groupe, tandis que l'ensemble des éléments nécessaires au projet final reste centralisé et versionné sur GitLab.

9 Difficultés rencontrées et limites

La mise en œuvre de ce projet a soulevé plusieurs difficultés, liées à la fois à la complexité technique de l'architecture demandée, au nombre important de services à intégrer et aux outils de virtualisation utilisés.

9.1 Complexité globale de l'architecture

L'un des principaux défis a été la gestion de la complexité globale du projet. La plateforme repose sur de nombreuses briques interdépendantes : système autonome, routage dynamique, accès box-like pour les particuliers, services d'entreprise (DNS, DHCP, VPN, VoIP, WEB), mécanismes de sécurité et QoS. Chaque modification locale (adressage, routage, filtrage) pouvait avoir des impacts sur l'ensemble du réseau, rendant le débogage parfois long et délicat. La nécessité de maintenir une vision globale cohérente a donc été un enjeu central tout au long du projet.

9.2 Difficultés liées à l'utilisation de GNS3

L'utilisation de GNS3, bien que très adaptée à un contexte pédagogique, s'est révélée exigeante dans la pratique. La compatibilité entre les différents environnements de travail (systèmes d'exploitation, versions de GNS3, configuration Docker locale) a parfois posé problème lors du travail collaboratif, notamment pour le partage ou la reprise de projets sur des machines différentes.

Par ailleurs, la gestion de la persistance des configurations dans GNS3 a constitué une difficulté importante. Les conteneurs Docker utilisés ne conservent pas nativement les modifications effectuées à l'exécution, ce qui peut entraîner des pertes de configuration lors des redémarrages. La mise en place de mécanismes de sauvegarde et de restauration fiables a donc nécessité un travail conséquent, afin de garantir la reproductibilité du projet et d'éviter toute configuration résiduelle incohérente.

9.3 Déploiement et débogage des services

Certains services se sont révélés plus complexes à déployer que prévu. En particulier, la mise en place de la VoIP a nécessité de nombreuses itérations, tant sur la configuration d'Asterisk que sur les aspects réseau (NAT, ports, DNS, accès Internet réel pour les tests). De manière générale, le débogage de services réseau dans un environnement virtualisé (GNS3 + Docker) demande une méthodologie rigoureuse, basée sur l'analyse des logs, les captures réseau et des tests progressifs.

9.4 Gestion de la persistance et de la reproductibilité

Au-delà des difficultés propres à GNS3, assurer une reproductibilité complète de la plateforme a constitué un enjeu majeur. La nécessité de pouvoir redémarrer l'ensemble de l'infrastructure sans perte de configuration a conduit à la mise en place de scripts dédiés. Bien que ces scripts aient permis de stabiliser le projet, leur développement et leur validation ont représenté un investissement important en temps et en effort.

9.5 Interconnexion inter-AS non finalisée

La principale limite fonctionnelle du projet concerne l'interconnexion inter-AS, qui n'a pas pu être mise en œuvre dans le temps imparti. Cette étape aurait permis de valider pleinement les scénarios d'échange entre groupes et l'accès aux sites secondaires hébergés sur d'autres AS. Bien que l'architecture ait été conçue pour accueillir cette interconnexion (routeur de bordure dédié, plan d'adressage cohérent et routage interne fonctionnel), son absence limite la validation expérimentale complète du scénario Internet simplifié.

9.6 Contraintes temporelles

Enfin, le facteur temps a constitué une contrainte importante. La richesse du cahier des charges, combinée à la volonté de produire une plateforme stable, fonctionnelle et documentée, a nécessité des arbitrages. Certaines fonctionnalités ont ainsi été simplifiées ou laissées à l'état de conception afin de garantir la qualité et la robustesse de l'ensemble déjà implémenté.

Malgré ces difficultés, les solutions mises en place et les choix réalisés ont permis d'aboutir à une infrastructure cohérente et opérationnelle, offrant une base solide pour des améliorations et extensions futures.

10 Perspectives et améliorations

Le projet a permis de mettre en place une architecture cohérente et plusieurs services fonctionnels (routage interne, DNS/DHCP, VPN, WEB, VoIP). Néanmoins, plusieurs axes d'amélioration peuvent être identifiés afin de renforcer la conformité au cahier des charges et de se rapprocher d'un scénario plus réaliste.

10.1 Correction et harmonisation du plan d'adressage

Une première amélioration concerne le plan d'adressage IP. Initialement, notre groupe (groupe 1) a utilisé une plage avec 1 au lieu de 0. Nous n'avions pas remarqué l'astérisque en fin de page du sujet, et avons interprété la règle comme *n* plutôt que *n–1*. En pratique, cette approximation n'a pas posé de problème durant le projet, dans la mesure où l'interconnexion inter-AS n'a pas été finalisée. Même en cas d'interconnexion, il aurait été possible de choisir un AS voisin utilisant une plage différente (groupe 2, 3, etc.). Une perspective d'amélioration consisterait donc à **corriger l'adressage sur l'ensemble de la plateforme** et à **redéfinir le plan d'adressage** afin d'être strictement conforme au cahier des charges et de faciliter une interconnexion future sans ambiguïté.

10.2 Finalisation de l'interconnexion inter-AS

L'interconnexion inter-AS reste la principale fonctionnalité non implémentée. Une amélioration prioritaire serait de mettre en place une interconnexion effective entre plusieurs projets, soit en regroupant les topologies dans un même environnement GNS3, soit en utilisant la connectivité de la machine hôte comme lien de transit. Une fois les liens établis, l'échange de routes pourrait rester simple (routes statiques).

10.3 Tests et validation fonctionnelle

De manière plus générale, le projet gagnerait à être accompagné d'une **campagne de tests plus systématique**. Bien que de nombreux tests ponctuels aient été réalisés (connectivité, DNS, VPN, web), une validation plus complète pourrait inclure :

— des scénarios de test bout-en-bout (client particulier vers services d'entreprise),
— des tests de charge et de concurrence (notamment pour la QoS),
— une validation systématique de tous les services déployés après redémarrage complet de la plateforme.

L'objectif serait de garantir la robustesse globale du réseau et de démontrer le bon fonctionnement de l'ensemble des briques dans des conditions proches d'une exploitation réelle.

10.4 Sécurité et automatisation

Enfin, certaines améliorations pourraient renforcer la qualité globale du projet, notamment :

— un **durcissement de la sécurité** (règles de filtrage plus strictes, segmentation plus fine),
— une **automatisation plus avancée** du déploiement et de la configuration (outils d'orchestration, scripts de tests automatiques).

Ces évolutions faciliteraient la maintenance, le débogage et la reproductibilité de l'infrastructure.

11 Conclusion

Ce projet d'interconnexion s'est révélé particulièrement ambitieux et exigeant, tant par le nombre de briques techniques à intégrer (AS, routage dynamique, services, sécurité, QoS, VPN, VoIP) que par la nécessité de maintenir une cohérence globale entre toutes les configurations. Malgré cette charge de travail importante, nous avons réussi à concevoir et déployer une plateforme réseau solide et réaliste, organisée autour d'un système autonome jouant le rôle de fournisseur de services et d'un réseau d'entreprise structuré sur un site principal (et un site secondaire prévu).

Nous avons mis en œuvre un plan d'adressage cohérent, un routage dynamique interne via `OSPF`, ainsi qu'un accès *box-like* pour les clients particuliers (DHCP/NAT). Côté entreprise, les services essentiels demandés ont été intégrés et validés : DNS interne, DHCP, VPN (accès distant employé et inter-sites), serveur web et service VoIP. La mise en place de la QoS a également permis de différencier le trafic entreprise du trafic particulier, conformément au cahier des charges. L'ensemble forme une infrastructure fonctionnelle, reproductible et suffisamment modulaire pour être étendue.

Pour assurer cette cohérence globale, l'organisation du projet a été un point clé : la centralisation des choix d'architecture, la structuration des configurations et la mise en place de scripts de sauvegarde/restauration ont permis de stabiliser une plateforme pourtant composée de nombreux équipements. Le pilotage général du projet (coordination des tâches, validation des choix techniques, maintien de la cohérence d'ensemble) a notamment contribué à la qualité finale de l'intégration.

Certaines limites subsistent néanmoins. L'interconnexion inter-AS, composante centrale pour simuler un Internet simplifié, n'a pas pu être finalisée dans le temps imparti, ce qui a empêché la validation expérimentale des scénarios

d'échanges entre groupes. De plus, une erreur d'interprétation du cahier des charges a conduit à utiliser une plage d'adressage décalée (utilisation de `120.0.16*1.0/20` au lieu de `120.0.16*0.0/20`). Cette approximation n'a pas eu d'impact bloquant dans notre contexte actuel, mais elle devra être corrigée pour une interconnexion complète.

En conclusion, au regard de l'ampleur du projet, le résultat obtenu est très satisfaisant : la majorité des fonctionnalités attendues a été implémentée, testée et documentée, tout en conservant une architecture claire et évolutive. Les perspectives identifiées (interconnexion inter-AS, correction de l'adressage, campagne de tests plus systématique) constituent des améliorations naturelles permettant d'aboutir à une plateforme pleinement interconnectée et encore plus proche d'un déploiement réel.

12 Répartition des tâches

1. Chef de projet : Rémi AIRIAU
2. Choix des outils et de l'architecture (GNS3 + Docker, images utilisées) : Rémi AIRIAU, Mano Dinnat Creusier, Marina Galstyan
3. Mise en place de l'environnement Docker (conteneurs, réseaux) : Rémi AIRIAU, Marina Galstyan
4. Conception de la topologie globale (AS, sites, clients) : Rémi AIRIAU
5. Plan d'adressage IP et gestion de la plage 120.0.16x.0/20 : Rémi AIRIAU, Hassan El Sahily
6. Routage interne dynamique (OSPF) : Hassan El Sahily, Mano Dinnat Creusier
7. DNS (AS / Entreprise) : Rémi AIRIAU, Marina Galstyan
8. DHCP (Entreprise / Box clients particuliers) : Rémi AIRIAU, Malcolm BERTAINA
9. Accès clients particuliers (box + NAT) : Malcolm BERTAINA
10. VPN (Employé ↔ Entreprise / Site ↔ Site) :Malcolm BERTAINA
11. Service VoIP : Rémi AIRIAU
12. Service applicatif (Web / autre) : Rémi AIRIAU
13. Interconnexion inter-AS (conception / mise en œuvre partielle) : Hassan El Sahily, Marina Galstyan (recherches, l'interconnexion n'a pas eu lieu)
14. Scripts et automatisation (sauvegarde / restauration des configurations) : Rémi AIRIAU
15. Qualité de service (QoS) : Hassan El Sahily
16. Rédaction du rapport : Marina Galstyan, Rémi AIRIAU, Malcolm BERTAINA, Mano Dinnat Creusier, Hassan El Sahily
17. Autre : Mano Dinnat Creusier