

Compte-rendu Projet Interconnexion

Aloïs MEURISSE, Cédric GIRONCEL, Aurélien RAKOTOARISON, Romain COSTI,
Yahya MAZOUARI, Mathéo LIN-TENG-SHEE, Yassine SALIM

Janvier 2026

1 Introduction

Ce document présente la mise en place d'une infrastructure réseau virtualisée simulant une interconnexion entre réseaux. L'ensemble repose sur Docker et FRRouting pour le routage dynamique OSPF.

2 Architecture Réseau

L'architecture est segmentée en trois zones distinctes :

- **Backbone OSPF** : Composé de Router-Border et Router-Internal.
- **LAN Entreprise** : Zone protégée par une Box Debian avec redirection de port (NAT) et DNS menteur.
- **LAN Particulier** : Zone domestique utilisant un adressage dynamique (DHCP).

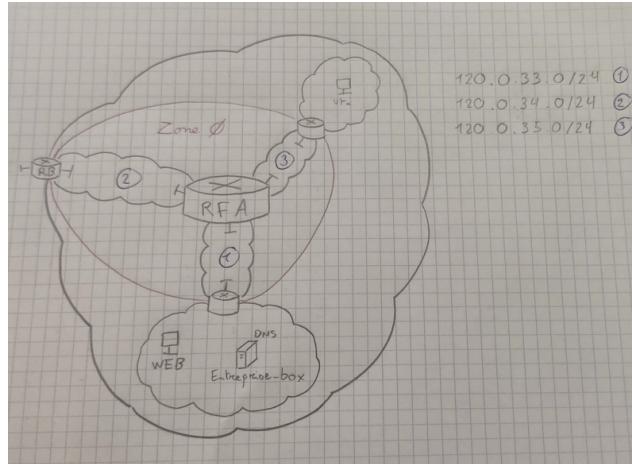


FIGURE 1 – Architecure du réseau

3 Environnement de Virtualisation Docker

3.1 Principe de Conteneurisation

L'utilisation de Docker permet de simuler une topologie réseau complexe sans la surcharge d'une virtualisation lourde (VM). Chaque composant (Routeur, Box, PC) fonctionne dans un conteneur isolé partageant le noyau de la machine hôte.

L'orchestration est réalisée via un fichier `docker-compose.yml` qui définit :

- **Les Services** : Images utilisées (Alpine, Debian, FRR).
- **Les Réseaux** : Déclaration des bridges virtuels et des sous-réseaux.
- **Les Volumes** : Montage des fichiers de configuration pour FRRouting.

3.2 Architecture des Dossiers

Pour assurer la persistance des configurations de routage, une structure de répertoires spécifique est utilisée pour monter les fichiers de configuration OSPF (`daemons`, `frr.conf`) dans les conteneurs.

4 Validation des Services (Tests Bash)

4.1 Vérification du DHCP (Côté Client)

Pour valider l'attribution dynamique, nous vérifions l'adresse IP du PC-Salon après le démarrage du service `udhcpc`.

```
1 # Commande exécutee sur l'hôte
2 docker exec -it PC-Salon ip addr show eth0
3
4 # Résultat attendu
5 2: eth0@if843: <BROADCAST,MULTICAST,UP,LOWER_UP> ...
6     inet 192.168.197.200/26 brd 192.168.197.255 scope global eth0
```

Listing 1 – Test de réception d'IP dynamique

4.2 Consultation des baux DHCP (Côté Serveur)

Sur la AS-Particulier-Box, nous vérifions que le serveur `dnsmasq` a bien enregistré le client.

```
1 docker exec -it Particulier-Box cat /var/lib/misc/dnsmasq.leases
2
3 # Résultat : Date d'expiration, MAC, IP et Nom du PC
4 1705928400 3a:e4:91:78:d5:58 192.168.197.200 PC-Salon 01:3a:e4:91...
```

Listing 2 – Vérification du bail sur la Box

Malheureusement nous n'avons pas réussi à faire fonctionner le DHCP le jour de la démonstration.

4.3 Test DNS et Connectivité HTTP

Le test final consiste à résoudre le nom de domaine de l'entreprise et à récupérer la page web depuis le réseau domestique.

```
1 # Test de resolution DNS
2 docker exec -it PC-Salon nslookup www.entreprise.com
3 # Reponse : Address: 120.0.98.3 (IP de la Box Entreprise)
4
5 # Recuperation de la page via le NAT
6 docker exec -it PC-Salon wget -qO- www.entreprise.com
7
8 # Resultat
9 <h1>BIENVENUE SUR LE SITE DE L ENTREPRISE</h1>
```

Listing 3 – Resolution DNS et Téléchargement HTTP

5 Configuration du NAT et de la Sécurité

La Box Entreprise dissimule le serveur web réel (192.168.197.131). Voici la règle de translation de destination (DNAT) appliquée :

```
1 # Redirection du trafic HTTP entrant vers le serveur interne
2 iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination
   192.168.197.131:80
3
4 # Masquage des adresses internes pour la sortie (SNAT)
5 iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Listing 4 – Configuration IPTables

6 Conclusion

Ce projet nous a permis de faire nos premiers pas sur Docker. Parallèlement à nos recherches et à notre apprentissage de cette plateforme, nous avons pu y appliquer ce que l'on a vu en TP. On a su mettre en place un routage dynamique (OSPF), configurer des serveurs web et dns, mis en place le NAT. Bien que nous n'ayons pas eu le temps ni l'occasion de nous interconnecter avec un autre réseau, nous avons tout de même préparé un routeur de bordure prévu à cet effet.