

# TP NE430 N°3 - OSPF

SONKO Mohamet | ROUGÉ Jean

## Intro

Ce TP nous fait découvrir le protocole ospf et son utilisation dans un réseau complexe.

Le schéma du réseau est fourni en fin de CR. La source est disponible ici

## Template

Ci-dessous, le script bash qui permet de générer nos configurations pour les routeurs.

```
#!/bin/bash

NPOP=$1
MACHINE_TYPE=$2

# Numéro de zone
Arr=(0 32 64 96 32)
AREA=${Arr[$NPOP]}

echo "conf terminal"
echo "routeur ospf"

# Routeur Backbone
if [[ $MACHINE_TYPE -eq 0 ]]; then
    echo "network 10.0.0.x/30 area 0"
    echo "area $AREA virtual_link 10.224.$AREA.25y"
    # Exigence 5
    echo "# area $AREA range 10.32.0.0/11"
# Router HQ
elif [[ $MACHINE_TYPE -eq 1 ]]; then
    echo "network 10.$AREA.0.0/11 area $AREA"
    # Exigence 3
    echo "# redistribute static"
# Routeur Point de Vente
elif [[ $MACHINE_TYPE -eq 2 ]]; then
    echo "network 10.$AREA.0.0/20 area $AREA"
fi

echo "network 10.224.$AREA.0/24 area $AREA"
echo "exit"

# Exigence 5 Routeur Backbone
if [[ $MACHINE_TYPE -eq 0 ]]; then
    echo "# interface eth1"
    echo "# ip ospf cost 1000"
    echo "# exit"
# Exigence 3 Routeur HQ
elif [[ $MACHINE_TYPE -eq 1 ]]; then
    #
    echo "# ip route a.b.c.0/24 Null0"
fi

echo "exit"
```

```
echo "show run"
```

Le scripts prends en entrée deux argument: - Le numéro du POP 1, 2, 3 ou 4 - Le type de machine -0→ Routeur Backbone -1→ Routeur Point de Vente -2→ Routeur HQ

A noter que par manque de temps, les commandes répondant aux exigence n'ont pas été testé, elles sont donc précédées d'un # pour ne pas être interprétée par bash.

De plus certaines commandes ne peuvent pas être généraliser et doivent être traitée au cas par cas. Le template contient donc des lettres à la place de certains numéros dans les addresses. `x ligne15; y ligne16; a,b,c ligne41`

## Configuration de base

On accède au menu de configuration ospf avec les commandes `configure terminal` et `routeur ospf`.

**Pour tous les routeurs:** On partage le réseau local dans l'aire du POP associé. `network 10.224.$AREA.0/24 area $AREA`

**Pour les routeurs de Backbone:** On partage le réseau qui relie deux routeurs de backbone dans l'aire de backbone. `network 10.0.0.x/30 area 0` On crée un lien virtuel entre les deux routeurs de backbone du POP. `area $AREA virtual_link 10.224.$AREA.25y`

**Pour les routeurs de HQ:** On partage les sous réseaux du HQ (englobés dans un /11) dans l'aire du POP. `network 10.$AREA.0.0/11 area $AREA`

**Pour les routeurs Point de vente:** On partage les sous réseaux des points de vente (englobés dans un /20) dans l'aire du POP. `network 10.$AREA.0.0/20 area $AREA`

## Exigences

### Exigence 1 :smiling\_imp:

Nous n'avons pas réussi à implémenter cette exigence, notre première idée à été de bloquer le trafic entre les routeur POP.Rx à partir du switch, mettre une règle pour que ces routeurs puissent uniquement communiquer avec les routeur BK et HQ. Mais il nous à été fait remarquer que cette solution ne fonctionnerait pas. On pourrait alors peut-être filtrer le trafic avec un pare-feu et `nftables` comme nous avons pu le faire l'année dernière.

### Exigence 2

Pour indiquer, via DNS, l'adresse des serveurs applicatifs des différents HQ, nous voulions modifier le fichier `/etc/resolve.conf` et indiquer l'adresse des différents HQ de notre architecture : `nameserver @IP_HQ` Les adresses de chaque HQ sont considérées comme déjà visible depuis chaque point de vente.

### Exigence 3

Sur chacun des routeurs on crée des routes statiques vers l'interface null (pour simuler une connection à un autre réseau) `ip route a.b.c.0/24 Null0` Puis on redistribue ces routes avec la commande `redistribute static`.

### Exigence 4

D'après nos tests, nous étions capables de ping toutes les machines du réseau sans configurations supplémentaires. Peut-être que la configuration des liens virtuels aurait du se trouver dans cette exigence.

### Exigence 5

Cette commande permet de limiter l'envoi des détails pour ne pas partager aux autres POPs les détails des réseaux en dessous des /11. `area $AREA range 10.32.0.0/11`

On rajoute un coût sur les interfaces de backbones pour ne pas privilégier ces routes. `interface eth1 ip ospf cost 1000`

## Schéma

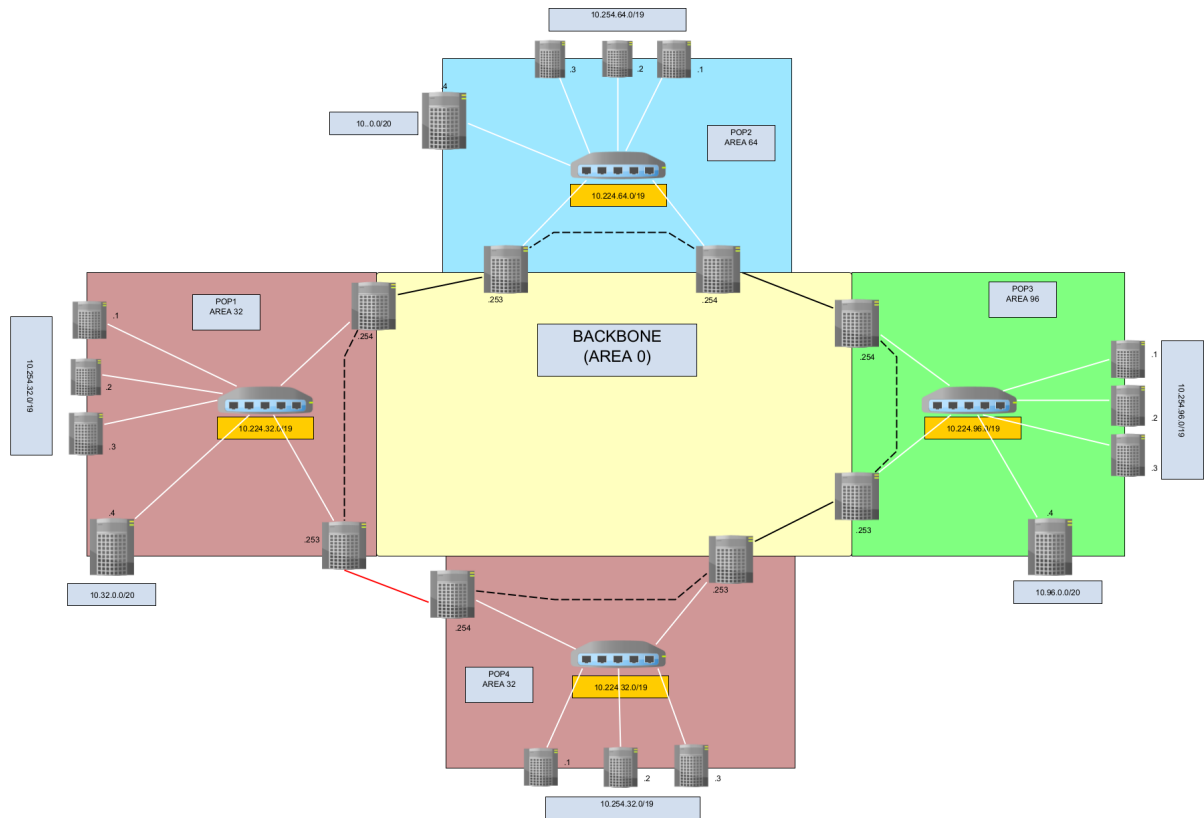


Figure 1: schéma