



Table des matières

1) Premier script : Gestion de la commande traceroute.....	1
2) Second script : Affichage de la cartographie.....	2

Les deux scripts sont à part et doivent être exécutés manuellement pour pouvoir avoir une cartographie à la fin.

En cours nous avons étudiée la commande traceroute (ce qu'elle fait, à quoi elle sert...) Après l'avoir étudiée on nous a demandé de créer un script qui automatisera cette commande pour pouvoir au final sortir une cartographie des routeurs traversés. J'ai donc créé deux scripts qui vont permettre de remplir les missions données.

1) Premier script : Gestion de la commande traceroute

Le premier script va gérer tout ce qui est la récupération d'IP et l'envoi dans un fichier .rte

Il est basé sur le TTL qui ira de 1 à 30

Dans ce script on a une boucle pour le TTL, une boucle pour l'utilisation des méthodes et deux boucles pour la récupération d'IP (une pour récupérer les 10 premières IP et l'autre pour récupérer toutes les autres)

Toutes les IP récupérées sont envoyées dans le fichier ip.rte qui sera ensuite géré par le deuxième script

Voici donc mon script :

```

GNU nano 2.9.3                                new_traceroute.sh
1  #!/bin/bash
2
3  #Le script commence par demander d'entrer une IP ou un nom de domaine.
4  #Ensuite on a la définition des variables comme méthodes ou ttl.
5  #À partir de là le script commence.
6  #Tout d'abord le script est basé sur le ttl. On va donc récupérer des IP tant que le ttl ne sera...
7  #...pas arrivé au bout.
8  #Le ttl s'incrémente de un à chaque IP récupérée
9  #On écrit la méthode utilisée pour tracer l'IP puis on fait la commande traceroute une par une.
10 #Les commandes traceroute des 10 premiers ttl récupérées sont testées. On les coupe pour ne...
11 #...garder que l'IP et on test si c'est une étoile ou une IP qui ressort de la commande.
12 #Si c'est une IP qui ressort on la garde et elle est envoyée dans le fichier .rte
13 #Ensuite pour tous les ttl supérieurs à 10 on fait la même chose mais lors de la coupure des...
14 #...champs on ne garde pas les mêmes que ceux des ttl inférieurs à 10 sinon les IP sont coupées.
15
16 rm ip.rte
17 echo "Entrez une IP ou un nom de domaine à tracer :"
18 read ip
19 methodes=(-I" "-U -p 443" "-U -p 53" "-U -p 123" "-T -p 443" "-T -p 53" "-T -p 123")
20 ttl=0
21 maxttl=30
22 while [ $ttl != $maxttl ]
23 do
24     ((++ttl))
25     for methode in $methodes
26     do
27         echo "Méthode utilisée : $methode"
28         commande="sudo traceroute $methode -n -f $ttl -m $ttl -N 1 -q 1 $ip"
29
30         if [ $ttl -ge 10 ]
31         then
32             testIP=$(($commande|cut -d " " -f 3|tail -n 1)
33             if [ "$testIP" != '*' ]
34             then
35                 $commande|cut -d " " -f 3,4|tail -n 1 >> ip.rte
36                 break
37             fi
38             else
39                 testIP=$(($commande|cut -d " " -f 4|tail -n 1)
40                 if [ "$testIP" != '*' ]
41                 then
42                     $commande|cut -d " " -f 4,5|tail -n 1 >> ip.rte
43                     break
44                 fi
45             fi
46         done
47     done
48

```

Ce script aurait pu être un peu amélioré si j'avais ajouté une boucle permettant de couper le TTL quand une IP récupérée est égale à l'IP de la destination finale. La boucle aurait été dans le même style que la suivante :

Si une des IP récupérée est égale à l'IP de destination
 Alors faire :
 Fin du programme

2) Second script : Affichage de la cartographie

Le second script lui va gérer l'affichage de la cartographie au format .pdf

Dans ce script d'abord on va définir une variable qui va permettre d'arrêter le script quand il n'y a plus d'IP dans le fichier ip.rte

Ensuite les IP sont récupérées une par une et sont mises à la suite entre « » pour que le fichiers .dot puisse être lu sans erreurs

les IP sont envoyées dans le fichier .dot et on termine par la compilation du fichier .dot et l'exécution du fichier .pdf

```
GNU nano 2.9.3                               exec.sh                               Modifié

#!/bin/bash

#On définit les variables
#On met le fichier .dot dans un format lisible
#Tout ce qui est dans le fichier ip.rte est récupéré et coupé pour ne garder que l'IP.
#Les IP sont mises entre " les unes derrières les autres avec une flèche entre les deux pour...
#...qu'elles puissent être interprétées dans le fichier .xdot
#On compile le fichier .xdot pour le transformer en fichier .pdf
#On exécute le fichier .pdf grâce à la commande atril

rm -rf carto.xdot
longueur=1
echo "digraph A {" >> carto.xdot
for ip in ip.rte
do
    echo $ip
    taille=$(wc -l $ip|cut -d " " -f 1)
    echo "Nombre d'IP dans le fichier : $taille"
    while [ $longueur -lt $taille ]
    do
        longueur1=$((longueur + 1))
        ipa=$(cat $ip|head -n $longueur1|tail -n 1)
        ipb=$(cat $ip|head -n $longueur|tail -n 1)
        ((longueur+=1))
        echo "\"$ipb\"->\"$ipa\" [label=\"$ip\"]" >> carto.xdot
    done
    longueur=1
done
echo } >> carto.xdot
dot -Tpdf carto.xdot -o carto.pdf
atril carto.pdf
```