



INSTITUT FRANCOPHONE POUR L'INNOVATION

UNIVERSITÉ NATIONAL DU VIETNAM, HANOI

TP1 de Conception et Architecture Reseaux

Groupe n°1

Professeur : Dr. NGUYEN HONG
QUANG

Table des matières

| | | |
|----------|--|-----------|
| 1 | Introduction | 2 |
| 2 | Interfaces réseaux | 2 |
| 2.1 | Liste des interfaces de notre machine | 2 |
| 2.2 | IP, adresse MAC, Masque sous-reseau | 3 |
| 2.3 | La table de routage | 3 |
| 2.4 | Nom de domaine et serveur de l'adresse 112.137.140.41 | 3 |
| 2.5 | Liste des routeurs | 4 |
| 2.6 | Serveurs de nom pour les domaines ftp.com.vn et ifi.edu.vn | 4 |
| 3 | Configuration d'une interface wifi | 4 |
| 4 | Analyse de trames | 4 |
| 4.1 | Analyse du protocole ARP | 4 |
| 5 | Fonctionnement de l'outil mtr | 6 |
| 5.1 | Générer un rapport MTR | 7 |
| 5.2 | Champs qui varient entre l'envoi des paquets : | 8 |
| 5.3 | lecture de rapport MTR | 8 |
| 5.4 | architecture réseau | 8 |
| 6 | Analyse détaillée du protocole TCP | 9 |
| 7 | Analyse détaillée du protocole TELNET | 12 |

1 Introduction

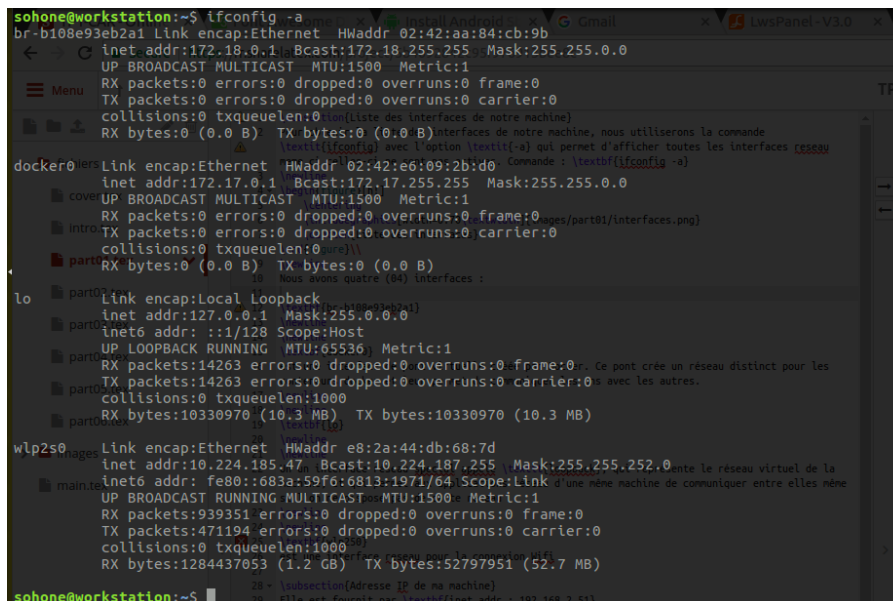
Dans le cadre de notre formation au module Conception et Architecture Réseau, il nous a été soumis un TP qui a pour objectifs de nous permettre de réaliser des tâches d'administration sur un poste de travail linux en utilisant des commandes mais aussi quelques outils pratique pour un administrateur système.

Ce présent rapport résume les différents tâches qui nous ont été assignées et est réparties en 6 sections. La deuxième concerne **les interfaces réseau**, la troisième traite de **la configuration d'une interface wifi**. En section 4 **l'analyse des captures de trames**, la cinquième section décrit **le fonctionnement de l'outil mtr**. Une analyse détaillée du protocole TCP a la section 6, la section 7 pour l'analyse du protocole TELNET et **la conclusion** en section 8

2 Interfaces réseaux

2.1 Liste des interfaces de notre machine

Pour obtenir la liste des interfaces de notre machine, nous utiliserons la commande *ifconfig* avec l'option *-a* qui permet d'afficher toutes les interfaces réseau même si celles-ci ne sont pas actives. Commande : **ifconfig -a**



```
sohonne@workstation:~$ ifconfig -a
br-b108e93eb2a1 Link encap:Ethernet HWaddr 02:42:aa:84:cb:9b
  inet addr:172.18.0.1 Bcast:172.18.255.255 Mask:255.255.0.0
  UP BROADCAST MULTICAST MTU:1500 Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

docker0 Link encap:Ethernet HWaddr 02:42:e6:09:2b:d0
  inet addr:172.17.0.1 Bcast:172.17.255.255 Mask:255.255.0.0
  UP BROADCAST MULTICAST MTU:1500 Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

lo Link encap:Local Loopback
  inet addr:127.0.0.1 Mask:255.0.0.0
  inet6 addr: ::1/128 Scope:Host
  UP LOOPBACK RUNNING MTU:65536 Metric:1
  RX packets:14263 errors:0 dropped:0 overruns:0 frame:0
  TX packets:14263 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:10330970 (10.3 MB) TX bytes:10330970 (10.3 MB)

wlp2s0 Link encap:Ethernet HWaddr e8:2a:44:db:68:7d
  inet addr:10.224.185.47 Bcast:10.224.187.255 Mask:255.255.252.0
  inet6 addr: fe80::683a:59f6:6813:1971/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
  RX packets:939351 errors:0 dropped:0 overruns:0 frame:0
  TX packets:471194 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:1284437053 (1.2 GB) TX bytes:52797951 (52.7 MB)
```

FIGURE 1 – Liste des interfaces

Nous avons quatre (04) interfaces :

br-b108e93eb2a1 : est une interface de pont virtuelle créée par VirtualBox.

docker0 : est une interface de pont virtuelle créée par Docker. Ce pont crée un réseau distinct pour les conteneurs docker et leur permet de communiquer les uns avec les autres.

lo : Une interface réseau spéciale appelée *loopback*, qui représente le réseau virtuel de la machine, et

qui permet aux applications réseau d'une même machine de communiquer entre elles même si l'on ne dispose pas de carte réseau

wlp2s0 est une interface réseau pour la connexion Wifi

Nous remarquons l'absence d'interface eth0, cela s'explique par le fait que le modèle de mon ordinateur n'a pas d'interface Ethernet.

2.2 IP, adresse MAC, Masque sous-réseau

Adresse IP : Elle est fournie par **inet addr** : **10.224.185.47**

Adresse MAC Wifi : Elle est fournie par **HWaddr** : **e8 :2a :44 :db :68 :7d**

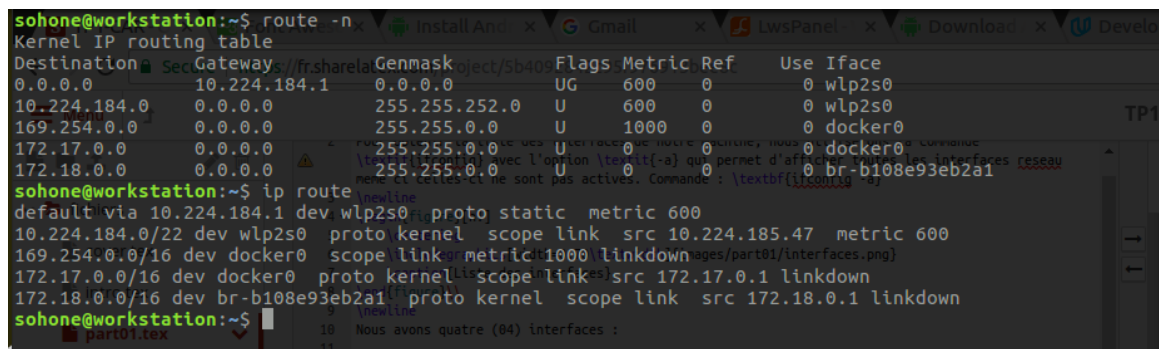
Masque sous-réseau : il est défini par **Mask** : **255.255.252.0**

2.3 La table de routage

Pour l'affichage de la table de routage, nous avons le choix entre trois commandes dont :

1. **ip** : **ip route**
2. **netstat** : **netstat -rn**
3. **route** : **route -n**

Notons que les deux commandes sont dépréciées au profit de la première commande, néanmoins, nous utiliserons les deux afin de vous montrer les outputs.



```
sohane@workstation:~$ route -n
Kernel IP routing table
Destination:Sec Gateway:Genmask:Flags:Metric:Ref:Use:Iface
0.0.0.010.224.184.10.0.0.0UG60000wlp2s0
10.224.184.00.0.0.0255.255.252.0U60000wlp2s0
169.254.0.00.0.0.0255.255.0.0U100000docker0
172.17.0.00.0.0.0255.255.0.0U00000docker0
172.18.0.00.0.0.0255.255.0.0U00000br-b108e93eb2a1

sohane@workstation:~$ ip route
default via 10.224.184.1 dev wlp2s0 proto static metric 600
10.224.184.0/22 dev wlp2s0 proto kernel scope link src 10.224.185.47 metric 600
169.254.0.0/16 dev docker0 scope link metric 1000 linkdown
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown
172.18.0.0/16 dev br-b108e93eb2a1 proto kernel scope link src 172.18.0.1 linkdown

sohane@workstation:~$
```

FIGURE 2 – Table de routage

La passerelle qui nous permet de sortir de notre réseau local a l'adresse : « 10.224.184.1 » connectée à l'interface wifi wlp2s0. Pour atteindre notre réseau « 10.224.184.0 » on passe par la route par défaut « 0.0.0.0 ».

2.4 Nom de domaine et serveur de l'adresse 112.137.140.41

Afin de pouvoir déterminer le nom de domaine et du serveur de l'adresse ci-dessous, nous avons les commandes **nslookup** et **dig**

Nous indique que le serveur de nom est a l'adresse **127.137.140.41** tandis que le nom de domaine n'a pu être fourni.


```

sohone@workstation:~$ dig NS ftp.com.vn
; <<>> DiG 9.10.3-P4-Ubuntu <<>> NS: ftp.com.vn
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 16044
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;, udp: 512
;; QUESTION SECTION:
;ftp.com.vn.
IN NS
ftp.com.vn. 3599 IN NS ns2.dns.net.vn
ftp.com.vn. 3599 IN NS ns1.dns.net.vn
;; Query time: 200 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Sun Jul 08 16:30:04 +07 2018
;; MSG SIZE rcvd: 83
> part01
sohone@workstation:~$ dig NS ifi.edu.vn
; <<>> DiG 9.10.3-P4-Ubuntu <<>> NS: ifi.edu.vn
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 15872
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;, udp: 512
;; QUESTION SECTION:
;ifi.edu.vn.
IN NS
ifi.edu.vn. 21599 IN NS dns.vnu.edu.vn.
;; Query time: 79 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Sun Jul 08 16:32:13 +07 2018
;; MSG SIZE rcvd: 61

```

FIGURE 5 – Nom de domaine et ServerName

```

sohone@workstation:~$ arp -a
logout.lan (10.224.184.1) at ac:86:74:26:13:12 [ether] on wlp2s0
sohone@workstation:~$

```

FIGURE 6 – Cache arp

Effectuons des « pings » sur l'adresse « IP » du réseau local qui n'est pas dans le cache ARP à l'aide de la commande : « ping 10.224.185.86 ». La figure ci-dessous montre le résultat cette commande.

la figure ci-dessous montre que notre adresse MAC en vois des message de diffusion aux machines qui sont dans le réseau pour leur demandé alors l'adresse physique de la machine qui a l'adresse IP« 10.224.185.86».On constate que c'est le message **request** du protocole ARP.Bien-que toute les machines reçoivent le message seul la machine qui a l'adresse Ip 10.224.185.86 répond a la requête en lui envoyant son adresse mac a l'émetteur de la requête.comme illustration la figure ci-dessous :

le protocole ARP nous permet de retrouver l'adresse physique d'une machine à partir de son adresse MAC. Pour ce faire un message request contenant l'adresse IP de la machine dont on recherche l'adresse physique est envoyé par la machine émettrice en Broadcast vers toutes les machines du réseau mais seule la machine dont l'adresse IP correspond à celle contenue dans la requête répond

```

sohone@workstation:~$ ping 10.224.184.17
PING 10.224.184.17 (10.224.184.17) 56(84) bytes of data:
From 10.224.185.47 icmp_seq=1 Destination Host Unreachable
From 10.224.185.47 icmp_seq=2 Destination Host Unreachable
From 10.224.185.47 icmp_seq=3 Destination Host Unreachable
From 10.224.185.47 icmp_seq=4 Destination Host Unreachable
From 10.224.185.47 icmp_seq=5 Destination Host Unreachable
From 10.224.185.47 icmp_seq=6 Destination Host Unreachable
From 10.224.185.47 icmp_seq=7 Destination Host Unreachable
From 10.224.185.47 icmp_seq=8 Destination Host Unreachable
From 10.224.185.47 icmp_seq=9 Destination Host Unreachable
From 10.224.185.47 icmp_seq=10 Destination Host Unreachable
From 10.224.185.47 icmp_seq=11 Destination Host Unreachable
From 10.224.185.47 icmp_seq=12 Destination Host Unreachable
From 10.224.185.47 icmp_seq=13 Destination Host Unreachable
From 10.224.185.47 icmp_seq=14 Destination Host Unreachable
From 10.224.185.47 icmp_seq=15 Destination Host Unreachable
From 10.224.185.47 icmp_seq=16 Destination Host Unreachable
From 10.224.185.47 icmp_seq=17 Destination Host Unreachable
From 10.224.185.47 icmp_seq=18 Destination Host Unreachable
From 10.224.185.47 icmp_seq=19 Destination Host Unreachable
From 10.224.185.47 icmp_seq=20 Destination Host Unreachable
From 10.224.185.47 icmp_seq=21 Destination Host Unreachable
From 10.224.185.47 icmp_seq=22 Destination Host Unreachable
From 10.224.185.47 icmp_seq=23 Destination Host Unreachable
From 10.224.185.47 icmp_seq=24 Destination Host Unreachable

```

FIGURE 7 – Pings autre machine du réseau

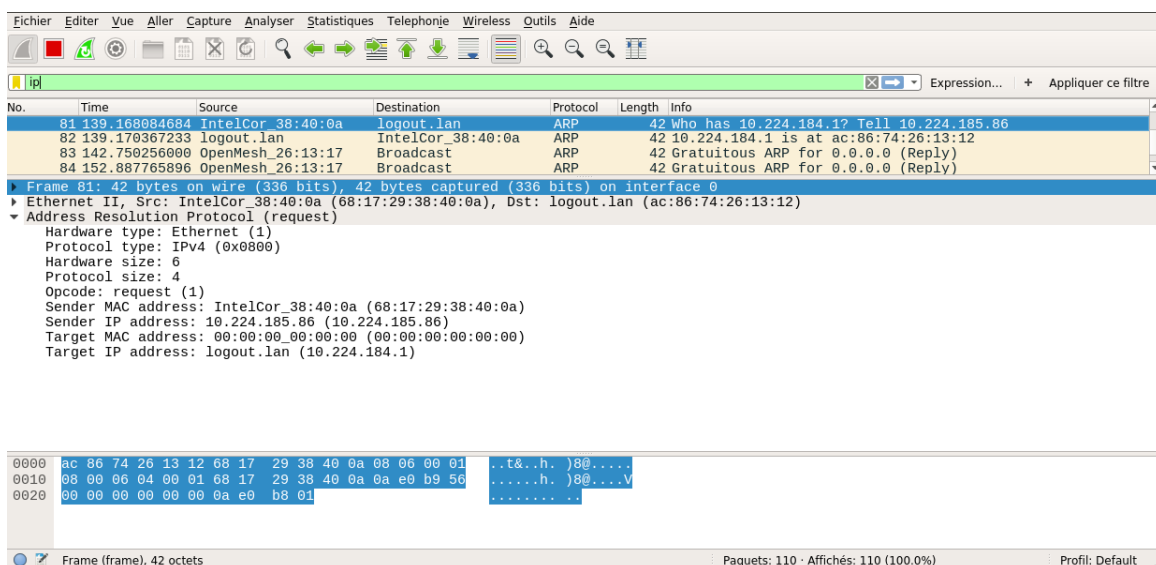


FIGURE 8 – Contenu de la trame ARP (request)

au message en envoyant son adresse MAC.

5 Fonctionnement de l'outil mtr

Dans cette partie nous allons traité du fonctionnement de l'outil **MTR**. MTR est une sorte de **traceroute** combiné avec ping. C'est un outil de surveillance réseau incomparable. Comme traceroute, MTR nous indique chaque bond effectué par les paquets pour arriver à destination et comme ping, il nous donne pour chaque bond le nombre de paquets perdus, la latence(*elle désigne le temps nécessaire à un paquet de données pour passer de la source à la destination à travers un réseau.*) et des données statistiques. Afin de vérifier my trace route nous allons lancer la commande **mtr** avec une adresse (www.vnpt.com.vn) dans le but d'observer en détail les différents paquets envoyé sur le réseau. mtr www.vnpt.com.vn comme résultat nous obtenons :

Mtr nous affiche alors en continu les résultats de ses envois de paquets avec statistiques détaillées.on

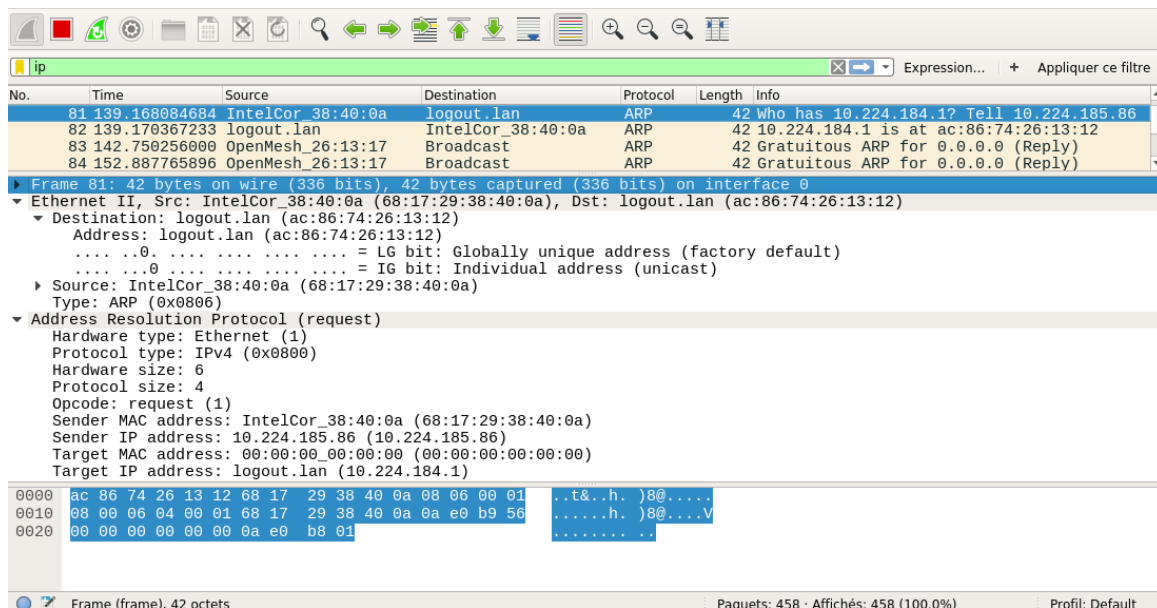


FIGURE 9 – Contenu de la trame ARP (request)

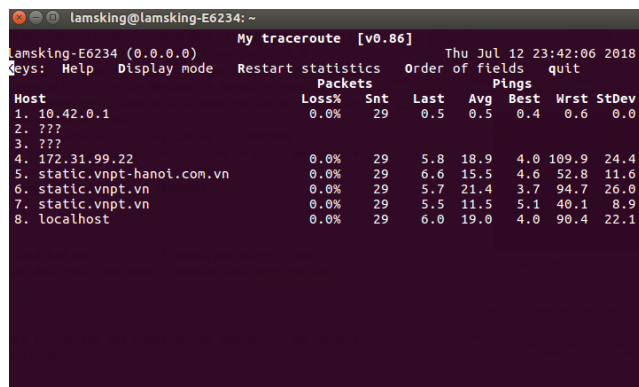


FIGURE 10 – Affichage des résultats avec l'adresse www.vnpt.com.vn

constate alors que les paquets passe par six routeur (6) avant d'arriver à destination. La colonne Loss(%) affiche le pourcentage de perte de paquets à chaque saut. La colonne **Snt** compte le nombre de paquets envoyés. L'option `-report` enverra 10 paquets sauf si spécifié avec `-report-cycles = [nombre-de-paquets]`, où `[nombre-de-paquets]` représente le nombre total de paquets que l'on veut envoyer à l'hôte distant. Les quatre colonnes suivantes Last, Avg, Best et Wrst sont toutes des mesures de latence en millisecondes. La dernière est la latence du dernier paquet envoyé, **Avg** est la latence moyenne de tous les paquets, tandis que **Best** et **Wrst** affichent le meilleur (le plus court) et le pire (le plus long) aller-retour d'un paquet vers cet hôte. Dans la plupart des cas, la colonne moyenne (moyenne) devrait être au centre de notre attention. La dernière colonne, **StDev**, fournit l'écart type des latences à chaque hôte. Plus l'écart type est élevé, plus la différence est grande entre les mesures de latence. L'écart type permet d'évaluer si la moyenne fournie représente le vrai centre de l'ensemble de données, ou a été faussée par une sorte de phénomène ou d'erreur de mesure. Si l'écart type est élevé, les mesures de latence sont incohérentes.

5.1 Générer un rapport MTR

Notons que mtr est un outil bidirectionnel car, fournir la route du trafic entre deux hôtes. La route empruntée entre deux points sur Internet peut varier énormément en fonction de l'emplacement et

des routeurs situés en amont. Pour cette raison, il est recommandé de collecter des rapports MTR dans les deux directions pour tous les hôtes rencontrant des problèmes de connectivité. Connecté depuis la machine suivante :

```
wlp2s0 : <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default
qlen 1000 link/ether cc:b0:da:b3:22:9d brd ff:ff:ff:ff:ff:ff inet 192.168.137.10/24 brd 192.168.137.255
scope global dynamic wlp2s0 valid_lft 604590sec preferred_lft 604590sec inet6 fe80::7b06:2cf9:24bb:64e6/64
scope link valid_lft forever preferred_lft forever
```

la figure qui suit est obtenu via la commande **mtr www.vnpt.com.vn** .

| Hostname | Loss | Snt | Last | Avg | Best | Worst | StDev |
|----------------------------|--------|-----|------|-----|------|-------|-------|
| DESKTOP-V2RL6KK.mshome.net | 0,0% | 31 | 7 | 29 | 2 | 137 | 48,19 |
| ??? | 100,0% | 31 | 0 | 0 | 0 | 0 | 0,00 |
| logout.lan | 3,3% | 30 | 9 | 24 | 3 | 119 | 39,74 |
| ??? | 100,0% | 30 | 0 | 0 | 0 | 0 | 0,00 |
| 172.31.99.22 | 0,0% | 30 | 6 | 28 | 4 | 159 | 42,61 |
| static.vnpt-hanoi.com.vn | 0,0% | 30 | 6 | 11 | 5 | 46 | 8,41 |
| static.vnpt.vn | 0,0% | 30 | 6 | 20 | 5 | 69 | 21,26 |
| static.vnpt.vn | 0,0% | 30 | 8 | 38 | 5 | 158 | 55,86 |
| localhost | 0,0% | 30 | 21 | 15 | 5 | 48 | 13,37 |

FIGURE 11 – affichage des résultats de la commande mtr avec l'adresse www.vnpt.com.vn

Chaque ligne numérotée de la sortie représente un saut. Les sauts sont les nœuds Internet que les paquets traverse pour se rendre à destination. Les noms des hôtes sont déterminés par des recherches DNS reverses.

mtr nous montre les différents équipement (leur adresses IP) entre la destination et notre machine. De plus il fait un rapport en temps réel de l'état de cette route à chaque paquet envoyé. Par fois, certain équipement (routeur , machine) vont rejeter l'ICMP , ce qui sera montré sur l'affichage pour « ??? ». Sinon, cela peut aussi être une problème avec la route emprunté par les paquets

5.2 Champs qui varient entre l'envoi des paquets :

- Snt : les paquets envoyés
- Last : temps de latence du dernier paquet envoyé
- Avg : la moyenne des latences
- StDev : l'écart type des latences à chaque hôte

5.3 lecture de rapport MTR

Si nous voulons omettre les recherches rDNS, vous pouvez utiliser l'option **-no-dns**, qui produit une sortie similaire à celle ci-dessous : **mtr -report -no-dns www.vnpt.com.vn**

5.4 architecture réseau

```

Start: Fri Jul 13 05:10:47 2018
HOST: debian
Loss% Snt Last Avg Best Wrst StDev
1|-- 192.168.137.1      0.0%  10  4.5  3.0  1.9  4.5  0.7
2|-- ???              100.0  10  0.0  0.0  0.0  0.0  0.0
3|-- 10.224.184.1      0.0%  10 123.7 17.7  3.5 123.7 37.4
4|-- ???              100.0  10  0.0  0.0  0.0  0.0  0.0
5|-- 172.31.99.22      0.0%  10 132.5 21.2  5.5 132.5 39.2
6|-- 123.25.27.177     0.0%  10 126.1 19.1  5.3 126.1 37.6
7|-- 123.29.5.41       0.0%  10 116.8 17.9  5.2 116.8 34.7
8|-- 113.171.33.42     0.0%  10   9.6  8.6  6.2  12.1  1.6
9|-- 123.31.40.181     0.0%  10   7.7  7.3  6.0  11.1  1.5

```

FIGURE 12 – affichage du rapport de la commande mtr avec l'adresse www.vnpt.com.vn

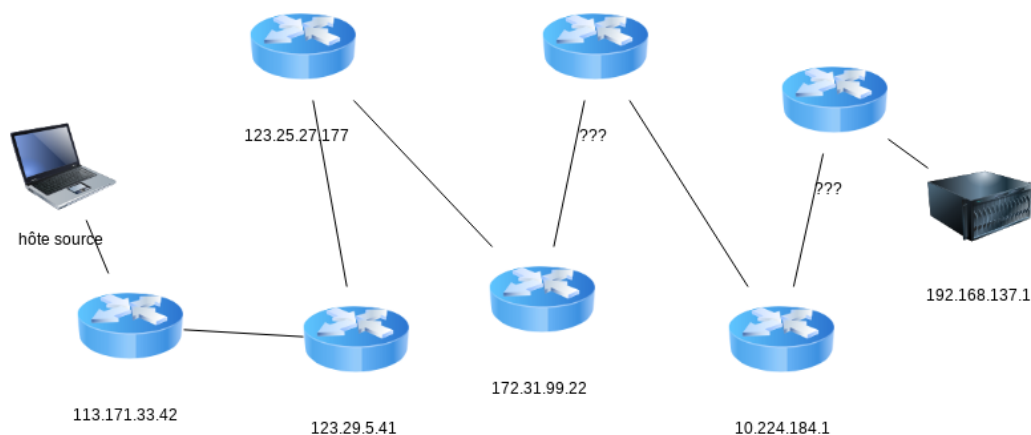


FIGURE 13 – architecture du réseau

6 Analyse détaillée du protocole TCP

TCPdump est, comme son nom peut l'indiquer, un analyseur de flux et de paquets réseau. Son rôle est donc de capturer tout ou partie d'un flux réseau transitant au travers une ou plusieurs interfaces d'une machine. L'analyse réseau est souvent utilisée dans l'administration système à des fins d'apprentissage et également à des fins de diagnostics. C'est intéressant, car on voit exactement le contenu de ce qui arrive et sort via le réseau. L'avantage de TCPdump est qu'il s'utilise en ligne de commande sous Linux au contraire de Wireshark qui possède une interface graphique.

Pour commencer nous allons lancer la capture avec tcpdump. Et pour cela nous allons utiliser la commande suivante : `(tcpdump -i wlp2s0 -w WS_user-guide-a4.pcap)` qui nous permettra de faire des captures. De ce fait on commence par charger la capture par le menu File > Open (on choisit le fichier .pcap). Une fois le fichier chargé, on peut voir que la fenêtre de Wireshark est, par défaut, divisée en 3 sections :

- la première affiche une liste des paquets IP capturés
- la seconde donne le détail du paquet IP sélectionné dans la première section
- la troisième affiche le contenu (en hexadécimal) du paquet IP sélectionné dans la première section

Selon notre capture, il peut être utile d'appliquer un filtre qui ne va afficher que certains paquets. Alors il est également possible d'effectuer ce filtrage lors de la capture mais pour ne pas avoir perdu certaines informations nous n'avons pas choisi ce cas.

L'image ci-dessous montre les résultats obtenus grâce à Wireshark avec l'application du filtre pour ne

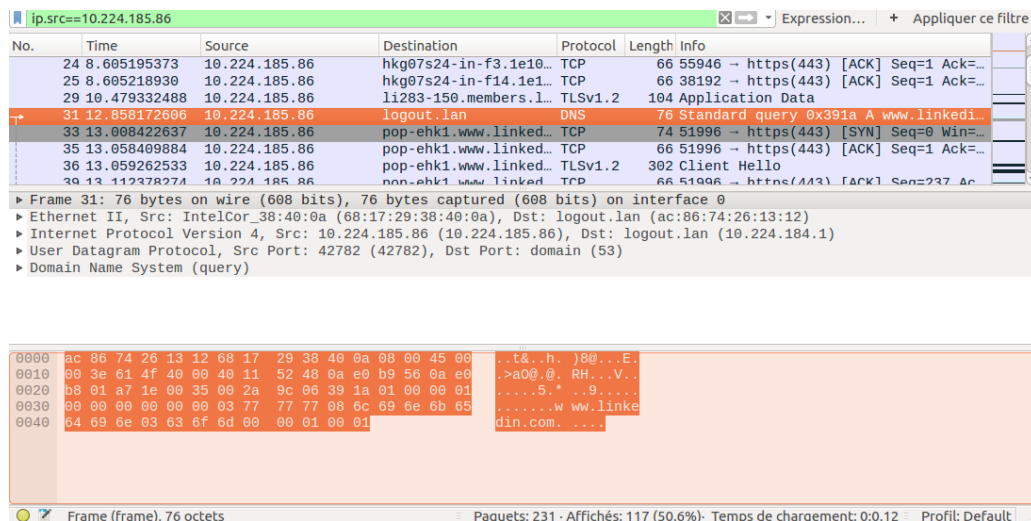


FIGURE 14 – Affichage des résultats avec l'adresse www.vnpt.com.vn

retenir que les données importante.

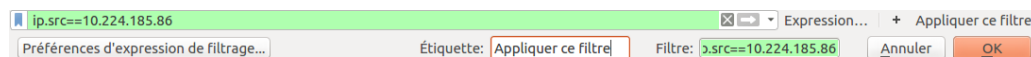


FIGURE 15 – application du filtre

il faut remarqué pour cette partie que si la zone qui est verte, est rouge cela sous-entend qu'il y a une ou plusieurs erreurs de syntaxe. on constate également que Le paquet en question porte le numéro 31, nous allons donc modifier légèrement le filtre pour n'afficher que les paquets dont le numéro est > à 30. ci-dessous l'image : la figure ci-dessous montre la liste complète des requêtes client/serveur

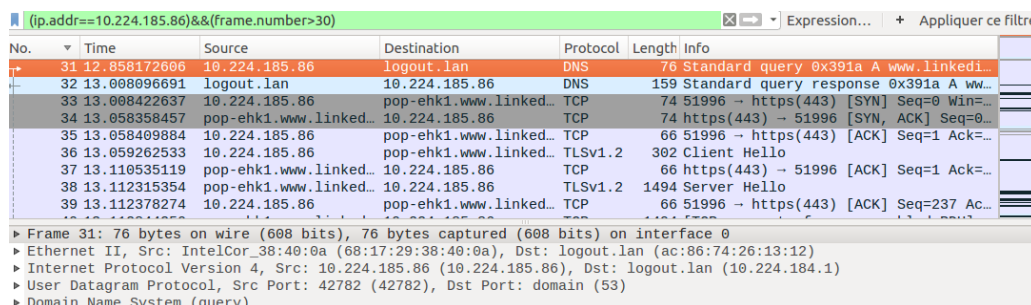


FIGURE 16 – filtrage des paquets supérieur à 30

| Ethernet · 2 | | IPv4 · 15 | IPv6 | TCP · 13 | UDP · 2 | | | |
|---------------|---|-----------|-------|---------------|-------------|---------------|-------------|--|
| Address A | Address B | Packets | Bytes | Packets A → B | Bytes A → B | Packets B → A | Bytes B → A | |
| 10.224.185.86 | pop-ehk1.www.linkedin.com | 28 | 8601 | 15 | 2734 | 13 | | |
| 10.224.185.86 | pagead46.l.doubleclick.net | 24 | 5044 | 13 | 3505 | 11 | | |
| 10.224.185.86 | tm-in-f188.1e100.net | 4 | 264 | 2 | 132 | 2 | | |
| 10.224.185.86 | a4.54.2ca9.ip4.static.sl-reverse.com | 9 | 843 | 6 | 510 | 3 | | |
| 10.224.185.86 | tm-in-f189.1e100.net | 16 | 1480 | 8 | 528 | 8 | | |
| 10.224.185.86 | edge-star-shv-02-hkg3.facebook.com | 12 | 1908 | 7 | 1041 | 5 | | |
| 10.224.185.86 | del01s08-in-f206.1e100.net | 4 | 264 | 2 | 132 | 2 | | |
| 10.224.185.86 | edge-star-z-mini-shv-02-hkg3.facebook.com | 13 | 2770 | 7 | 2015 | 6 | | |
| 10.224.185.86 | li283-150.members.linode.com | 6 | 540 | 2 | 208 | 4 | | |
| 10.224.185.86 | docs.google.com | 27 | 5447 | 15 | 2603 | 12 | | |
| 10.224.185.86 | ssl.gstatic.com | 33 | 7613 | 18 | 2224 | 15 | | |
| 10.224.185.86 | hkg07s24-in-f14.1e100.net | 2 | 132 | 1 | 66 | 1 | | |
| 10.224.185.86 | hkg07s24-in-f3.1e100.net | 2 | 132 | 1 | 66 | 1 | | |
| 10.224.185.86 | 239.255.255.250 | 2 | 418 | 2 | 418 | 0 | | |

FIGURE 17 – liste détaillé des requêtes client/serveur

- Connexion entre les machine Dans le but d’analyser les paquets échangés lors de l’utilisation du protocole TCP,nous nous sommes servi de cette commande afin de pouvoir lire que l’inter-
face qui nous intéresse qui est le WIFI. cette commande est « tcpdump -i wlp2s0 port http » et
ensuite dans un autre terminale nous exécutons la commande suivante en vue de télécharger
le fichier (WS_user-guide-a4.pdf); « wget http ://fad.ifi.edu.vn/ififad/file.php/28/documents/ASR_chap2_179.pdf »

Nous obtenons des resultats suivant montrant la connexion entre la machine et le server et
qui s’est effectuer en trois principales etapes a savoir :

- L’envoi d’un premier paquet au server(SYN)
- le server qui répond a son tour via un paquet (SYN+ACK) prouvant qu’il a reçu ce dernier
d’où l’acceptation d’une connexion,
- enfin la machine répond par un message (ACK) au server en vue de l’établissement de la
connexion.

```
97368033,nop,wscale 7], length 0
01:35:27.772889 IP 10.224.185.86.56700 > 112.137.140.42.http: Flags [..], ack 1,
win 229, options [nop,nop,TS val 97368037 ecr 51218341], length 0
01:35:27.772996 IP 10.224.185.86.56700 > 112.137.140.42.http: Flags [P..], seq 1:
191, ack 1, win 229, options [nop,nop,TS val 97368037 ecr 51218341], length 190:
HTTP: GET /ififad/file.php/28/documents/WS_user-guide-a4.pdf HTTP/1.1
01:35:27.776576 IP 112.137.140.42.http > 10.224.185.86.56700: Flags [..], ack 191
, win 54, options [nop,nop,TS val 51218342 ecr 97368037], length 0
01:35:28.086488 IP 112.137.140.42.http > 10.224.185.86.56700: Flags [P..], seq 1:
757, ack 191, win 54, options [nop,nop,TS val 51218419 ecr 97368037], length 756
```

FIGURE 18 – Séquence de connexion entre deux machines via le protocole TCP(tcpdump)

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|-------------|-----------------|-----------------|----------|--------|---|
| 35 | 5.826118733 | 10.224.185.86 | docs.google.com | TCP | 74 | 32888 → https(443) [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_P... |
| 36 | 5.851918677 | docs.google.com | 10.224.185.86 | TCP | 74 | https(443) → 32888 [SYN, ACK] Seq=0 Ack=1 Win=60192 Len=0 MSS=... |
| 37 | 5.851996609 | 10.224.185.86 | docs.google.com | TCP | 66 | 32888 → https(443) [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=420... |
| 38 | 5.852317811 | 10.224.185.86 | docs.google.com | TLSv1.2 | 583 | Client Hello |

FIGURE 19 – Séquence de connexion entre deux machines via le protocole TCP(wireshark)

- la phase de transfert de fichier L’image qui suit montre le transfert de fichier entre la machine
et le serveur via le GET. Si le serveur reçoit la requête alors elle renvoi un message ACK en
retour pour confirmer la réception du paquet puis enclenche le transfert du fichier.
- La phase de déconnexion
Une fois le transfert achevé, le client émet un paquet avec le champ « FIN,ACK » pour signifier
la fin de la connexion au serveur. Le serveur va de ce fait émettre également un paquet «
FIN,ACK » pour confirmation. Enfin le client envoie un paquet avec le champ « ACK » et la
connexion entre les deux machines se finis.

A présent montrons sur le diagramme ci-dessous les échanges entre la machine et le serveur quand
le protocole TCP qui a été utilisé.

```

root@lamsking-E6234: /home/lamsking
s [nop,nop,TS val 51218471 ecr 97368553], length 1428: HTTP
01:35:28.297287 IP 112.137.140.42.http > 10.224.185.86.56700: Flags [.], seq 7897:9325, ack 434, win 62, option
s [nop,nop,TS val 51218471 ecr 97368553], length 1428: HTTP
01:35:28.297338 IP 10.224.185.86.56700 > 112.137.140.42.http: Flags [.], ack 9325, win 374, options [nop,nop,TS
val 97368561 ecr 51218471], length 0
01:35:28.298409 IP 112.137.140.42.http > 10.224.185.86.56700: Flags [.], seq 9325:10753, ack 434, win 62, optio
ns [nop,nop,TS val 51218471 ecr 97368553], length 1428: HTTP
01:35:28.298451 IP 112.137.140.42.http > 10.224.185.86.56700: Flags [P.], seq 10753:11731, ack 434, win 62, opt
ions [nop,nop,TS val 51218471 ecr 97368554], length 978: HTTP
01:35:28.298478 IP 10.224.185.86.56700 > 112.137.140.42.http: Flags [.], ack 11731, win 419, options [nop,nop,T
S val 97368562 ecr 51218471], length 0
01:35:28.302829 IP 112.137.140.42.http > 10.224.185.86.56700: Flags [P.], seq 11731:11736, ack 434, win 62, opt
ions [nop,nop,TS val 51218473 ecr 97368554], length 5: HTTP
01:35:28.303781 IP 10.224.185.86.56700 > 112.137.140.42.http: Flags [F.], seq 434, ack 11736, win 419, options
[nop,nop,TS val 97368567 ecr 51218473], length 0
01:35:28.307052 IP 112.137.140.42.http > 10.224.185.86.56700: Flags [F.], seq 11736, ack 435, win 62, options [
nop,nop,TS val 51218474 ecr 97368567], length 0
01:35:28.307104 IP 10.224.185.86.56700 > 112.137.140.42.http: Flags [.], ack 11737, win 419, options [nop,nop,T
S val 97368571 ecr 51218474], length 0

```

FIGURE 20 – Séquence de connexion entre deux machines via le protocole TCP(tcpdump)

| | | | | | |
|-------|-----------------|---------------|------------------------|------|---|
| 34313 | 2227.9982939... | 10.224.185.86 | fad.ifi.edu.vn | HTTP | 256 GET /ififad/file.php/28/documents/WS_user-guide-a4.pdf HTTP/1.1 |
| 34315 | 2227.3307248... | 10.224.185.86 | fad.ifi.edu.vn | HTTP | 822 HTTP/1.1 303 See Other (text/html) |
| 34317 | 2227.3310614... | 10.224.185.86 | fad.ifi.edu.vn | HTTP | 309 GET /ififad/login/index.php HTTP/1.1 |
| 34329 | 2227.5579368... | 10.224.185.86 | fad.ifi.edu.vn | HTTP | 1049 HTTP/1.1 200 OK (text/html) |
| 36657 | 2575.1323575... | 10.224.185.86 | fad.ifi.edu.vn | HTTP | 256 GET /ififad/file.php/28/documents/WS_user-guide-a4.pdf HTTP/1.1 |
| 36661 | 2575.4573539... | 10.224.185.86 | fad.ifi.edu.vn | HTTP | 822 HTTP/1.1 303 See Other (text/html) |
| 36663 | 2575.4577866... | 10.224.185.86 | fad.ifi.edu.vn | HTTP | 309 GET /ififad/login/index.php HTTP/1.1 |
| 36679 | 2575.6772620... | 10.224.185.86 | fad.ifi.edu.vn | HTTP | 71 HTTP/1.1 200 OK (text/html) |
| 40601 | 2972.9592136... | 10.224.185.86 | a652.dscb.akamai.net | HTTP | 480 GET /MHEwbzBNMEswSTAJBgUrDgMCGGUABBRReAhtobFzTvhaRmVeJ38QUchY9AwQU... |
| 40603 | 2972.9638769... | 10.224.185.86 | a652.dscb.akamai.net | OCSP | 1190 Response |
| 43071 | 2983.0664613... | 10.224.185.86 | ghs.google.com | HTTP | 584 GET /2015/02/installation-et-configuration-telnet-linux.html HTTP... |
| 43100 | 2983.9841475... | 10.224.185.86 | googleapis.l.goog... | HTTP | 605 GET /css?family=Montserrat:400,700 Roboto:400,700,500,700italic,5... |
| 43103 | 2983.9853098... | 10.224.185.86 | googleapis.l.goog... | HTTP | 523 GET /css?family=Poppins:400,500,600,700 HTTP/1.1 |
| 43122 | 2984.0208998... | 10.224.185.86 | cds.j3z9t3p6.hwcdn... | HTTP | 535 GET /font-awesome/4.6.3/css/font-awesome.min.css HTTP/1.1 |
| 43136 | 2984.0308696... | 10.224.185.86 | googleapis.l.goog... | HTTP | 86 HTTP/1.1 200 OK (text/css) |
| 43144 | 2984.0334316... | 10.224.185.86 | googleapis.l.goog... | HTTP | 86 HTTP/1.1 200 OK (text/css) |
| 43217 | 2984.0801592... | 10.224.185.86 | cds.j3z9t3p6.hwcdn... | HTTP | 1016 HTTP/1.1 200 OK (text/css) |
| 43256 | 2984.1008986... | 10.224.185.86 | ghs.google.com | HTTP | 586 HTTP/1.1 200 OK (text/html) |
| 43418 | 2984.3642812... | 10.224.185.86 | ghs.google.com | HTTP | 696 GET /2015/02/installation-et-configuration-telnet-linux.html HTTP... |
| 43421 | 2984.3901179... | 10.224.185.86 | gstaticadssl.l.goog... | HTTP | 618 GET /s/montserrat/v12/JTUSjIgl_i6t8kCHKm459Wlhyw.woff2 HTTP/1.1 |
| 43465 | 2984.4410716... | 10.224.185.86 | gstaticadssl.l.goog... | HTTP | 719 HTTP/1.1 200 OK (font/woff2) |
| 43481 | 2984.4567598... | 10.224.185.86 | gstaticadssl.l.goog... | HTTP | 622 GET /s/montserrat/v12/JTUSjIgl_i6t8kCHKm45_dJE3gnD_g.woff2 HTTP/1... |
| 43482 | 2984.4568380... | 10.224.185.86 | pagead46.l.doublecl... | HTTP | 507 GET /pagead/js/adsbygoogle.js HTTP/1.1 |

FIGURE 21 – Séquence de connexion entre deux machines via le protocole TCP(wireshark)

| | | | | | |
|-------|-----------------|---------------|----------------|-----|--|
| 34331 | 2227.5588465... | 10.224.185.86 | fad.ifi.edu.vn | TCP | 66 57270 -> http(80) [FIN, ACK] Seq=434 Ack=11736 Win=53632 Len=0 TSv... |
| 34332 | 2227.5620378... | 10.224.185.86 | fad.ifi.edu.vn | TCP | 66 http(80) -> 57270 [FIN, ACK] Seq=11736 Ack=435 Win=7936 Len=0 TSv... |
| 34333 | 2227.5620850... | 10.224.185.86 | fad.ifi.edu.vn | TCP | 66 57270 -> http(80) [ACK] Seq=435 Ack=11737 Win=53632 Len=0 TSval=10... |

FIGURE 22 – Séquence de déconnexion entre deux machines via le protocole TCP(wireshark)

```

03:36:35.780925 IP 10.224.185.86.56614 > 112.137.140.42.http: Flags [R], seq 3952029164, win 0, length 0
03:36:35.780925 IP static.vnpt.vn.http > 10.224.185.86.56614: Flags [.], ack 415, win 235, options [nop,
nop,TS val 1279601287 ecr 667566723], length 0
03:36:35.780925 IP static.vnpt.vn.http > 10.224.185.86.56614: Flags [.], ack 415, win 235, options [nop,
nop,TS val 667611964 ecr 1279556093], length 0
03:37:21.088385 IP 10.224.185.86.56614 > static.vnpt.vn.http: Flags [.], ack 1126, win 246, options [nop
nop,TS val 1279646597 ecr 667611964], length 0
03:37:21.093315 IP static.vnpt.vn.http > 10.224.185.86.56614: Flags [R], seq 3952029164, win 0, length 0

```

FIGURE 23 – Séquence de déconnexion entre deux machines via le protocole TCP(tcpdump)

7 Analyse détaillée du protocole TELNET

Dans cette partie sera question de présenté le protocole TELNET.Ce protocole permet le transfert des informations des utilisateurs sur le réseau. De prime abord nous allons nous connecté a server TELNET sur des machines de notre réseau local avec la commande « telnet 192.168.137.52 » ensuite identifions nous en utilisant le compte « lamsking » avec le mot de passe « s6dxlame ».ci-dessous la figure. Avec les images qui suit on remarque que le port source (sur notre machine) est : 50244 et le port de destination (sur le serveur) est : 23. Nous trouvons le mot de passe en analysant la capture des trames obtenues de wireshark. En effet, comme nous l'avons fait ci-dessus, lorsqu'une machine fait un telnet sur une autre machine, il est obligé de taper son login et son mot de passe.De ce fait nous constatons que la capture va enregistré toutes actions d'où on peut voir en nette le mot de passe qui est « s6dxlame »

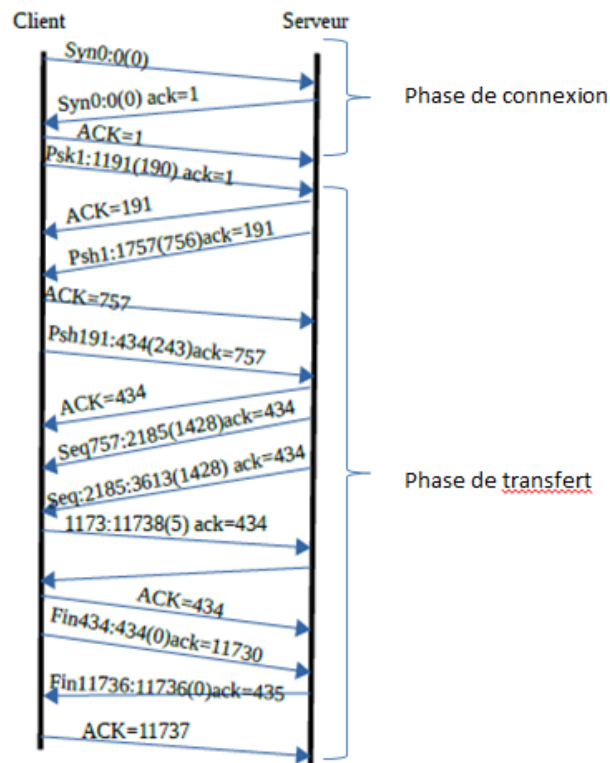


FIGURE 24 – Diagramme des échanges entre le client et le serveur pour le protocole TCP

```

saheed@saheed:~$ telnet 192.168.137.52
Trying 192.168.137.52...
Connected to 192.168.137.52.
Escape character is '^]'.
Ubuntu 16.04.4 LTS
lamsking-E6234 login: lamsking
Password:
Last login: Mon Jan 22 21:27:59 +07 2018 on tty1
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.15.0-24-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

23 paquets peuvent être mis à jour.
23 mises à jour de sécurité.

```

FIGURE 25 – Connexion au serveur telnet situé sur la machine d'adresse 192.168.137.52

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------|-----------------|-----------------|----------|--------|-----------------|
| 95 | 78.405483352 | 192.168.137.183 | 192.168.137.52 | TELNET | 93 | Telnet Data ... |
| 97 | 78.548311329 | 192.168.137.52 | 192.168.137.183 | TELNET | 78 | Telnet Data ... |
| 99 | 78.550488790 | 192.168.137.52 | 192.168.137.183 | TELNET | 105 | Telnet Data ... |
| 101 | 78.550653372 | 192.168.137.183 | 192.168.137.52 | TELNET | 149 | Telnet Data ... |
| 103 | 78.553339792 | 192.168.137.52 | 192.168.137.183 | TELNET | 69 | Telnet Data ... |
| 104 | 78.553454544 | 192.168.137.183 | 192.168.137.52 | TELNET | 69 | Telnet Data ... |
| 106 | 78.642911721 | 192.168.137.52 | 192.168.137.183 | TELNET | 69 | Telnet Data ... |
| 107 | 78.643060067 | 192.168.137.183 | 192.168.137.52 | TELNET | 69 | Telnet Data ... |
| 108 | 78.646336906 | 192.168.137.52 | 192.168.137.183 | TELNET | 86 | Telnet Data ... |
| 110 | 78.739996706 | 192.168.137.52 | 192.168.137.183 | TELNET | 88 | Telnet Data ... |

▶ Frame 95: 93 bytes on wire (744 bits), 93 bytes captured (744 bits) on interface 0
 ▶ Ethernet II, Src: IntelCor_6d:aa:b5 (c4:d9:87:6d:aa:b5), Dst: IntelCor_38:40:0a (68:17:29:38:40:0a)
 ▶ Internet Protocol Version 4, Src: 192.168.137.183, Dst: 192.168.137.52
 ▶ Transmission Control Protocol, Src Port: 50244, Dst Port: 23, Seq: 1, Ack: 1, Len: 27
 ▼ Telnet
 ▶ Do Suppress Go Ahead

FIGURE 26 – Identification du port source et du port destination

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|---------------|-----------------|-----------------|----------|--------|-----------------|
| 201 | 96.370524656 | 192.168.137.52 | 192.168.137.183 | TELNET | 67 | Telnet Data ... |
| 203 | 96.551298188 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 204 | 96.573548799 | 192.168.137.52 | 192.168.137.183 | TELNET | 67 | Telnet Data ... |
| 206 | 96.938262559 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 207 | 96.998583397 | 192.168.137.52 | 192.168.137.183 | TELNET | 67 | Telnet Data ... |
| 209 | 97.934163423 | 192.168.137.183 | 192.168.137.52 | TELNET | 68 | Telnet Data ... |
| 210 | 98.015685355 | 192.168.137.52 | 192.168.137.183 | TELNET | 68 | Telnet Data ... |
| 212 | 98.017177167 | 192.168.137.52 | 192.168.137.183 | TELNET | 76 | Telnet Data ... |
| 214 | 99.994928937 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 217 | 103.672938862 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |

▶ Frame 212: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
 ▶ Ethernet II, Src: HonHaiP_63:ff:82 (1c:3e:84:63:ff:82), Dst: IntelCor_6d:aa:b5 (c4:d9:87:6d:aa:b5)
 ▶ Internet Protocol Version 4, Src: 192.168.137.52, Dst: 192.168.137.183
 ▶ Transmission Control Protocol, Src Port: 23, Dst Port: 50244, Seq: 118, Ack: 131, Len: 10
 ▼ Telnet
 Data: Password:

```

0000 c4 d9 87 6d aa b5 1c 3e 84 63 ff 82 08 00 45 10  ...m...> .c...E
0010 00 3e 01 ce 40 00 40 06 a4 9f c0 a8 89 34 c0 a8  ->...@.@...4...
0020 89 b7 00 17 c4 44 61 2a a2 f2 36 67 ad 7a 80 18  ...Da* ..6g.z...
0030 00 e3 ae 3e 00 00 01 01 08 0a 15 1f 26 c8 9d b3  ...>...&...
0040 c5 8e 50 61 73 73 77 6f 72 64 3a 20          ..Passwo rd:
  
```

FIGURE 27 – Champ DATA avec la valeur « password »

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|---------------|-----------------|-----------------|----------|--------|-----------------|
| 210 | 98.015685355 | 192.168.137.52 | 192.168.137.183 | TELNET | 68 | Telnet Data ... |
| 212 | 98.017177167 | 192.168.137.52 | 192.168.137.183 | TELNET | 76 | Telnet Data ... |
| 214 | 99.994928937 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 217 | 103.672938862 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 219 | 104.292869692 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 223 | 104.500879079 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 225 | 105.341543718 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 228 | 105.757754206 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |

▶ Frame 214: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0
 ▶ Ethernet II, Src: IntelCor_6d:aa:b5 (c4:d9:87:6d:aa:b5), Dst: IntelCor_38:40:0a (68:17:29:38:40:0a)
 ▶ Internet Protocol Version 4, Src: 192.168.137.183, Dst: 192.168.137.52
 ▶ Transmission Control Protocol, Src Port: 50244, Dst Port: 23, Seq: 131, Ack: 128, Len: 1
 ▼ Telnet
 Data: s

FIGURE 28 – Affichage de la lettre «s»du mot de passe

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|---------------|-----------------|-----------------|----------|--------|-----------------|
| 210 | 98.015685355 | 192.168.137.52 | 192.168.137.183 | TELNET | 68 | Telnet Data ... |
| 212 | 98.017177167 | 192.168.137.52 | 192.168.137.183 | TELNET | 76 | Telnet Data ... |
| 214 | 99.994928937 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 217 | 103.672938862 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 219 | 104.292869692 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 223 | 104.500879079 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 225 | 105.341543718 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 228 | 105.757754206 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |

▶ Frame 217: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0
 ▶ Ethernet II, Src: IntelCor_6d:aa:b5 (c4:d9:87:6d:aa:b5), Dst: IntelCor_38:40:0a (68:17:29:38:40:0a)
 ▶ Internet Protocol Version 4, Src: 192.168.137.183, Dst: 192.168.137.52
 ▶ Transmission Control Protocol, Src Port: 50244, Dst Port: 23, Seq: 132, Ack: 128, Len: 1
 ▼ Telnet
 Data: 6

FIGURE 29 – Affichage de la lettre «6»du mot de passe

| telnet | | | | | | |
|---|---------------|-----------------|-----------------|----------|--------|-----------------|
| No. | Time | Source | Destination | Protocol | Length | Info |
| 210 | 98.015685355 | 192.168.137.52 | 192.168.137.183 | TELNET | 68 | Telnet Data ... |
| 212 | 98.017177167 | 192.168.137.52 | 192.168.137.183 | TELNET | 76 | Telnet Data ... |
| 214 | 99.994928937 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 217 | 103.672938862 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 219 | 104.292869692 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 223 | 104.500879079 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 225 | 105.341543718 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 228 | 105.757754206 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| ▶ Frame 219: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0 ▶ Ethernet II, Src: IntelCor_6d:aa:b5 (c4:d9:87:6d:aa:b5), Dst: IntelCor_38:40:0a (68:17:29:38:40:0a) ▶ Internet Protocol Version 4, Src: 192.168.137.183, Dst: 192.168.137.52 ▶ Transmission Control Protocol, Src Port: 50244, Dst Port: 23, Seq: 133, Ack: 128, Len: 1 ▼ Telnet Data: d | | | | | | |

FIGURE 30 – Affichage de la lettre «d» du mot de passe

| telnet | | | | | | |
|---|---------------|-----------------|-----------------|----------|--------|----------------|
| No. | Time | Source | Destination | Protocol | Length | Info |
| 210 | 98.015685355 | 192.168.137.52 | 192.168.137.183 | TELNET | 68 | Telnet Data .. |
| 212 | 98.017177167 | 192.168.137.52 | 192.168.137.183 | TELNET | 76 | Telnet Data .. |
| 214 | 99.994928937 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data .. |
| 217 | 103.672938862 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data .. |
| 219 | 104.292869692 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data .. |
| 223 | 104.500879079 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data .. |
| 225 | 105.341543718 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data .. |
| 228 | 105.757754206 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data .. |
| ▶ Frame 223: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0 ▶ Ethernet II, Src: IntelCor_6d:aa:b5 (c4:d9:87:6d:aa:b5), Dst: IntelCor_38:40:0a (68:17:29:38:40:0a) ▶ Internet Protocol Version 4, Src: 192.168.137.183, Dst: 192.168.137.52 ▶ Transmission Control Protocol, Src Port: 50244, Dst Port: 23, Seq: 134, Ack: 128, Len: 1 ▼ Telnet Data: x | | | | | | |

FIGURE 31 – Affichage de la lettre «x» du mot de passe

| telnet | | | | | | |
|---|---------------|-----------------|-----------------|----------|--------|-----------------|
| No. | Time | Source | Destination | Protocol | Length | Info |
| 210 | 98.015685355 | 192.168.137.52 | 192.168.137.183 | TELNET | 68 | Telnet Data ... |
| 212 | 98.017177167 | 192.168.137.52 | 192.168.137.183 | TELNET | 76 | Telnet Data ... |
| 214 | 99.994928937 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 217 | 103.672938862 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 219 | 104.292869692 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 223 | 104.500879079 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 225 | 105.341543718 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 228 | 105.757754206 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| ▶ Frame 225: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0 ▶ Ethernet II, Src: IntelCor_6d:aa:b5 (c4:d9:87:6d:aa:b5), Dst: IntelCor_38:40:0a (68:17:29:38:40:0a) ▶ Internet Protocol Version 4, Src: 192.168.137.183, Dst: 192.168.137.52 ▶ Transmission Control Protocol, Src Port: 50244, Dst Port: 23, Seq: 135, Ack: 128, Len: 1 ▼ Telnet Data: l | | | | | | |

FIGURE 32 – Affichage de la lettre «l» du mot de passe

| telne | | | | | | |
|---|---------------|-----------------|-----------------|----------|--------|-----------------|
| No. | Time | Source | Destination | Protocol | Length | Info |
| 210 | 98.015685355 | 192.168.137.52 | 192.168.137.183 | TELNET | 68 | Telnet Data ... |
| 212 | 98.017177167 | 192.168.137.52 | 192.168.137.183 | TELNET | 76 | Telnet Data ... |
| 214 | 99.994928937 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 217 | 103.672938862 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 219 | 104.292869692 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 223 | 104.500879079 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 225 | 105.341543718 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 228 | 105.757754206 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 230 | 106.532077110 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 232 | 106.977084945 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| ▶ Frame 228: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0 | | | | | | |
| ▶ Ethernet II, Src: IntelCor_6d:aa:b5 (c4:d9:87:6d:aa:b5), Dst: IntelCor_38:40:0a (68:17:29:38:40:0a) | | | | | | |
| ▶ Internet Protocol Version 4, Src: 192.168.137.183, Dst: 192.168.137.52 | | | | | | |
| ▶ Transmission Control Protocol, Src Port: 50244, Dst Port: 23, Seq: 136, Ack: 128, Len: 1 | | | | | | |
| ▼ Telnet | | | | | | |
| Data: a | | | | | | |

FIGURE 33 – Affichage de la lettre «a»

| telne | | | | | | |
|---|---------------|-----------------|-----------------|----------|--------|-------------|
| No. | Time | Source | Destination | Protocol | Length | Info |
| 210 | 98.015685355 | 192.168.137.52 | 192.168.137.183 | TELNET | 68 | Telnet Data |
| 212 | 98.017177167 | 192.168.137.52 | 192.168.137.183 | TELNET | 76 | Telnet Data |
| 214 | 99.994928937 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data |
| 217 | 103.672938862 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data |
| 219 | 104.292869692 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data |
| 223 | 104.500879079 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data |
| 225 | 105.341543718 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data |
| 228 | 105.757754206 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data |
| 230 | 106.532077110 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data |
| 232 | 106.977084945 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data |
| ▶ Frame 230: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0 | | | | | | |
| ▶ Ethernet II, Src: IntelCor_6d:aa:b5 (c4:d9:87:6d:aa:b5), Dst: IntelCor_38:40:0a (68:17:29:38:40:0a) | | | | | | |
| ▶ Internet Protocol Version 4, Src: 192.168.137.183, Dst: 192.168.137.52 | | | | | | |
| ▶ Transmission Control Protocol, Src Port: 50244, Dst Port: 23, Seq: 137, Ack: 128, Len: 1 | | | | | | |
| ▼ Telnet | | | | | | |
| Data: m | | | | | | |

FIGURE 34 – Affichage de la lettre «m» du mot de passe

| telne | | | | | | |
|---|---------------|-----------------|-----------------|----------|--------|-----------------|
| No. | Time | Source | Destination | Protocol | Length | Info |
| 225 | 105.341543718 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 228 | 105.757754206 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 230 | 106.532077110 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 232 | 106.977084945 | 192.168.137.183 | 192.168.137.52 | TELNET | 67 | Telnet Data ... |
| 234 | 107.618791422 | 192.168.137.183 | 192.168.137.52 | TELNET | 68 | Telnet Data ... |
| 236 | 107.621763896 | 192.168.137.52 | 192.168.137.183 | TELNET | 68 | Telnet Data ... |
| 238 | 107.767885673 | 192.168.137.52 | 192.168.137.183 | TELNET | 116 | Telnet Data ... |
| 240 | 108.313164315 | 192.168.137.52 | 192.168.137.183 | TELNET | 357 | Telnet Data ... |
| 242 | 109.492809427 | 192.168.137.52 | 192.168.137.183 | TELNET | 150 | Telnet Data ... |
| ▶ Frame 232: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0 | | | | | | |
| ▶ Ethernet II, Src: IntelCor_6d:aa:b5 (c4:d9:87:6d:aa:b5), Dst: IntelCor_38:40:0a (68:17:29:38:40:0a) | | | | | | |
| ▶ Internet Protocol Version 4, Src: 192.168.137.183, Dst: 192.168.137.52 | | | | | | |
| ▶ Transmission Control Protocol, Src Port: 50244, Dst Port: 23, Seq: 138, Ack: 128, Len: 1 | | | | | | |
| ▼ Telnet | | | | | | |
| Data: e | | | | | | |

FIGURE 35 – Affichage de la lettre «e» du mot de passe

8 Conclusion

Au regard de tous ce qui précède, nous avons pu capitaliser les connaissances issues des cours et de nos recherches personnelles en compétences d'administration et de configuration de poste de travail sous linux . Fort de ces acquis nous pourront apprendre la configuration et la sécurisation d'un petit réseau local qui nous conférerait les compétences d'administrateur linux.