



**RAPPORT DU TP1
CONCEPTION ET ARCHITECTURE DE RESEAUX**

Par :

**Dao Thuy Hong
FOTSING SIKADIE Gervais Sertilange
NKUBA KASANDA Lievin**

Supervisé par :
Nguyen Hong Quang

Hanoï, Octobre 2016

Table des matières

Tables des matières	1
Introduction	2
Première partie : Outils pour configuration	3
Deuxième partie : Outils de la capture des trames	9
Analyse des protocoles à l'aide des outils pour la capture des trames	9
Analyse des routes suivies par les paquets (l'outil mtr)	11
Analyse du protocole TCP	14
Conclusion	19
Références	20

Introduction

Le concept de réseau est aujourd'hui largement entré dans le vocabulaire des sciences sociales. En histoire, l'introduction du vocabulaire des réseaux a souvent été liée à des démarches situées à une échelle « micro » et travaillant à mettre en évidence l'agence individuelle. Depuis les années 1990, une analyse de réseaux plus formalisée a fait des apparitions épisodiques, et inégales selon les domaines linguistiques, dans d'autres travaux historiques fondés au contraire sur des observations systématiques à une échelle macro. Après 30 ans d'une intégration de la catégorie à la démarche historique, un véritable savoir-faire historien émerge autour des questionnements auxquels elle est associée et des méthodologies qu'elle implique. Cependant, si l'analyse de réseaux a déjà largement fait la preuve de son intérêt dans certains domaines spécifiques de l'histoire, il est de coutume pour chaque étudiant d'être en mesure de comprendre leurs principes généraux mais aussi de savoir les configurer et les analyser, le tout dans l'environnement UNIX car c'est le plus utilisé dans le monde professionnel et académique.

L'objectif est d'acquérir les aptitudes précédemment énumérées et plus précisément à connaître et savoir utiliser les commandes de base de UNIX pour configurer et tester les connexions réseau, savoir configurer en réseau un poste de travail sous Linux sans faire recours aux outils graphiques, savoir analyser les protocoles de communication à l'aide des programmes pour la capture des trames.

Notre travail comprend deux parties, la première parlera des outils pour configuration et la deuxième des outils de la capture des trames.

Première partie : Outils pour configuration

1) Liste des interfaces sur la machine

En tapant la commande : **ifconfig -a** on peut trouver les interfaces suivantes :

```
helenrene@HelenRene: ~
helenrene@HelenRene:~$ ifconfig -a
enp1s0    Link encap:Ethernet  HWaddr ec:f4:bb:99:22:a8
UP BROADCAST MULTICAST  MTU:1500  Metric:1
Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
Octets reçus:0 (0.0 B) Octets transmis:0 (0.0 B)

lo        Link encap:Boucle locale
inet adr:127.0.0.1  Masque:255.0.0.0
adr inet6: ::1/128 Scope:Hôte
UP LOOPBACK RUNNING  MTU:65536  Metric:1
Packets reçus:772 erreurs:0 :0 overruns:0 frame:0
TX packets:772 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1
Octets reçus:74784 (74.7 KB) Octets transmis:74784 (74.7 KB)

wlp2s0    Link encap:Ethernet  HWaddr 34:23:87:e3:78:03
inet adr:10.229.46.35 Bcast:10.229.47.255 Masque:255.255.252.0
adr inet6: fe80::9c8c:ceac:14e0:aaac/64 Scope:Lien
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
Packets reçus:7500 erreurs:0 :0 overruns:0 frame:5140
TX packets:5715 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 lg file transmission:1000
```

- **lo** *Link encap:Boucle locale(réseaux filaire)*

lo pour la boucle locale

- **wlp2s0** *Link encap:Ethernet HWaddr 34:23:87:e3:78:03(réseau wifi)*

pour le WIFI. Elle est connectée à un réseau local via l'adresse

IP:10.229.46.35 et a pour adresse MAC : 34:23:87:e3:78:03

- **enp1s0** *Link encap:Ethernet HWaddr ec:f4:bb:99:22:a8*

pour l'Ethernet. Cette interface n'est connectée à aucun réseau d'où

l'absence d'adresse IP. Elle a pour adresse MAC : ec:f4:bb:99:22:a8

2) Adresse IP de votre machine

- *réseau wi-fi* : **inet adr: 10.229.46.35 Bcast:10.229.47.255 Masque:255.255.252.0**

3) Adresse MAC de la carte réseau

Link encap:Ethernet **HWaddr 34:23:87:e3:78:03**

4) l'adresse et le masque de votre réseau

Cette adresse est une adresse morcelée. 6 bits ont été pris sur l'identifiant machine pour augmenter le nombre de sous réseaux et par conséquent augmenter le nombre de machines à adresser. L'adresse IP est : 10.229.46.35 en binaire cela donne :

00001010.11100101.00101110.00100011



L'adresse réseau est obtenue en mettant tous les bits de la partie machine à 0 et ceux de la partie réseau intact ce qui donne : 00001010.11100101.00101100.00000000 = 10.229.44.0

Le masque de réseau est obtenue en mettant les bits de la partie réseau à 1 et ceux de la partie machine à 0 ce qui donne : 11111111.11111111.11111000.00000000 = 255.255.252.0

adresse réseau: 10.229.44.0 *Masque:* 255.255.252.0

5) la table de routage de votre machine

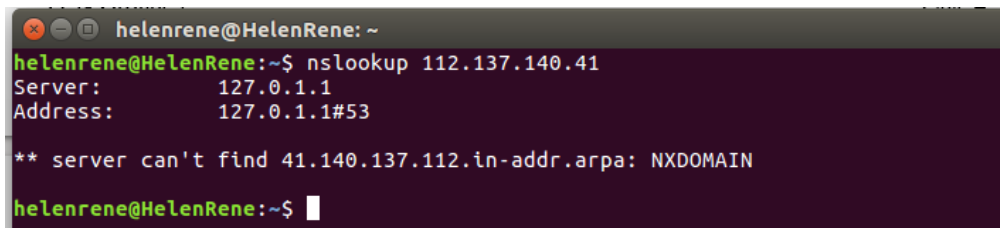
Afin d'accéder à la table de routage de notre machine nous avons tapé la commande

« route -n ». Le résultat obtenu est présenté comme suit :

```
helenrene@HelenRene: ~
helenrene@HelenRene:~$ route -n
Table de routage IP du noyau
Destination    Passerelle      Genmask          Indic Metric Ref    Use Iface
0.0.0.0         10.229.44.1     0.0.0.0          UG        600    0      0 wlp2s0
10.229.44.0     0.0.0.0         255.255.252.0    U         600    0      0 wlp2s0
169.254.0.0     0.0.0.0         255.255.0.0      U         1000   0      0 wlp2s0
helenrene@HelenRene:~$
```

Nous retenons de la table de routage que pour sortir de notre réseau local, vers n'importe qu'elle destination, nous devons prendre par la passerelle qui a pour adresse : « 10.229.44.1 » connectée à l'interface wlp2s0. Aussi, les paquets en direction de notre réseau « 10.229.44.0 » passeront par la route par défaut « 0.0.0.0 ».

- 6) Nom de la machine d'adresse IP 112.137.140.41 son domaine et le serveur de nom de son domaine. Cette commande ne renvoie aucun résultat.



```
helenrene@HelenRene: ~
helenrene@HelenRene:~$ nslookup 112.137.140.41
Server:      127.0.1.1
Address:     127.0.1.1#53

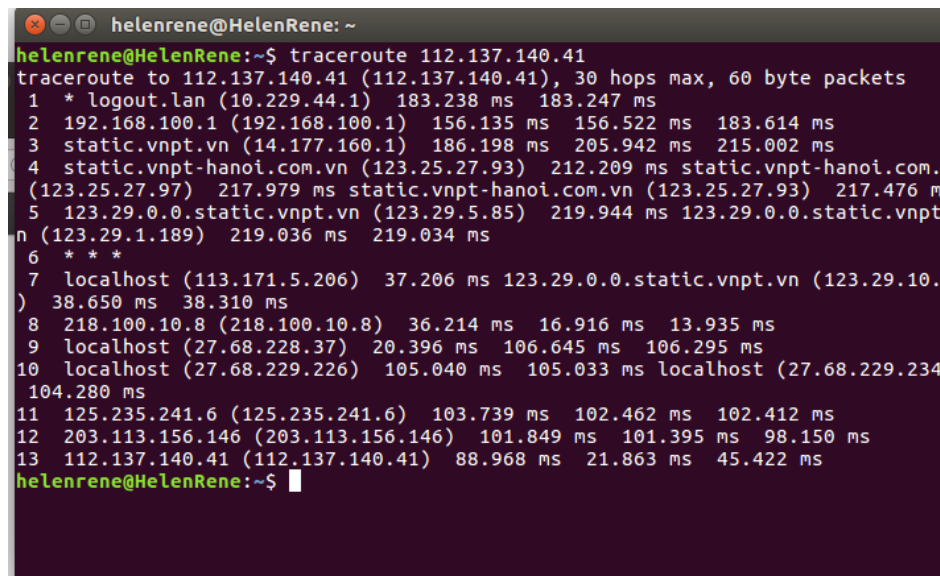
** server can't find 41.140.137.112.in-addr.arpa: NXDOMAIN

helenrene@HelenRene:~$
```

Illustration 3: commande pour rechercher le domaine et le nom de serveur de ce domaine

- 7) Liste des routeurs par lesquels doivent passer les datagrammes avant d'arriver à 112.137.140.41

La liste des routeurs traversés par les datagrammes entre notre machine et la machine d'adresse « 112.137.140.41 », comme présentée à la figure suivante, nous est fournie par la commande « *tracroute 112.137.140.41* »



```
helenrene@HelenRene: ~
helenrene@HelenRene:~$ traceroute 112.137.140.41
traceroute to 112.137.140.41 (112.137.140.41), 30 hops max, 60 byte packets
 1 * logout.lan (10.229.44.1) 183.238 ms 183.247 ms
 2 192.168.100.1 (192.168.100.1) 156.135 ms 156.522 ms 183.614 ms
 3 static.vnpt.vn (14.177.160.1) 186.198 ms 205.942 ms 215.002 ms
 4 static.vnpt-hanoi.com.vn (123.25.27.93) 212.209 ms static.vnpt-hanoi.com.
  (123.25.27.97) 217.979 ms static.vnpt-hanoi.com.vn (123.25.27.93) 217.476 m
 5 123.29.0.0.static.vnpt.vn (123.29.5.85) 219.944 ms 123.29.0.0.static.vnpt
 n (123.29.1.189) 219.036 ms 219.034 ms
 6 * * *
 7 localhost (113.171.5.206) 37.206 ms 123.29.0.0.static.vnpt.vn (123.29.10.
 ) 38.650 ms 38.310 ms
 8 218.100.10.8 (218.100.10.8) 36.214 ms 16.916 ms 13.935 ms
 9 localhost (27.68.228.37) 20.396 ms 106.645 ms 106.295 ms
10 localhost (27.68.229.226) 105.040 ms 105.033 ms localhost (27.68.229.234
 104.280 ms
11 125.235.241.6 (125.235.241.6) 103.739 ms 102.462 ms 102.412 ms
12 203.113.156.146 (203.113.156.146) 101.849 ms 101.395 ms 98.150 ms
13 112.137.140.41 (112.137.140.41) 88.968 ms 21.863 ms 45.422 ms
helenrene@HelenRene:~$
```

Illustration 4: routeurs traversé pour atteindre l'adresse 112.137.140.41

Nous constatons que nos datagrammes ont traversés treize (13) routeurs avant d'atteindre la machine d'adresse « 112.137.140.41 ».

- 8) Afin de connaître les serveurs de nom des domaines « *fpt.com.vn* » et « *ifi.edu.vn* » nous avons utilisé les commandes « *dig NS fpt.com.vn* » et « *dig NS ifi.edu.vn* ».

Les résultats obtenus sont décrits aux figures 5 et 6.

Nous y observons que les serveurs de nom du domaine « *fpt.com.vn* » sont *dns2.fpt.vn* (210.245.0.10) et *dns1.fpt.vn* (210.245.0.131) tandis que le serveur de nom du domaine « *ifi.edu.vn* » est *dns.vnu.edu.vn* (203.113.130.221)

```
helenrene@HelenRene:~$ dig NS fpt.com.vn

; <<>> DiG 9.10.3-P4-Ubuntu <<>> NS fpt.com.vn
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32092
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;fpt.com.vn.                IN      NS

;; ANSWER SECTION:
fpt.com.vn.                 3600    IN      NS      dns1.fpt.vn.
fpt.com.vn.                 3600    IN      NS      dns2.fpt.vn.

;; Query time: 652 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Mon Oct 24 11:48:54 ICT 2016
;; MSG SIZE rcvd: 81

helenrene@HelenRene:~$
```

Illustration 5: Informations sur le serveur de nom du domaine « *fpt.com.vn* »

```
helenrene@HelenRene:~$ 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: 34795
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;ifi.edu.vn.                IN      NS

;; ANSWER SECTION:
ifi.edu.vn.                 16793   IN      NS      dns.vnu.edu.vn.

;; Query time: 775 msec
;; SERVER: 127.0.1.1#53(127.0.1.1)
;; WHEN: Mon Oct 24 11:52:29 ICT 2016
;; MSG SIZE rcvd: 61

helenrene@HelenRene:~$
```

Illustration 6: Informations sur le serveur de nom du domaine

9) Configurer une interface wifi sans interface graphique

Pour configurer une interface wifi sous Linux sans avoir recours à des outils graphiques, nous disposons de deux moyens : soit nous utilisons des commandes, soit nous modifions le fichier « /etc/network/interfaces ».

-Ligne de commande

La commande « ifconfig » nous permet de modifier directement la configuration d'une interface wifi sous linux. Par exemple, si nous voulons attribuer l'adresse « 192.168.1.114 » à notre interface wifi, il nous faut taper la commande :

« sudo ifconfig wlp2s0 192.168.1.114 netmask 255.255.255.0 ».

La figure suivante donne la nouvelle adresse ip après l'exécution de cette commande :

```
helenrene@HelenRene:~$ ifconfig
enp1s0  Link encap:Ethernet  HWaddr ec:f4:bb:99:22:a8
        UP BROADCAST MULTICAST  MTU:1500  Metric:1
        Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
        TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:1000
        Octets reçus:0 (0.0 B) Octets transmis:0 (0.0 B)

lo      Link encap:Boucle locale
        inet adr:127.0.0.1  Masque:255.0.0.0
        adr inet6: ::1/128 Scope:Hôte
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        Packets reçus:2416 erreurs:0 :0 overruns:0 frame:0
        TX packets:2416 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:1
        Octets reçus:254620 (254.6 KB) Octets transmis:254620 (254.6 KB)

wlp2s0  Link encap:Ethernet  HWaddr 34:23:87:e3:78:03
        inet adr:192.168.1.114  Bcast:192.168.1.255  Masque:255.255.255.0
        adr inet6: fe80::a365:2f90:436d:127d/64 Scope:Lien
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        Packets reçus:36373 erreurs:0 :0 overruns:0 frame:41923
        TX packets:25264 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:1000
        Octets reçus:34390466 (34.3 MB) Octets transmis:3504001 (3.5 MB)
        Interruption:17

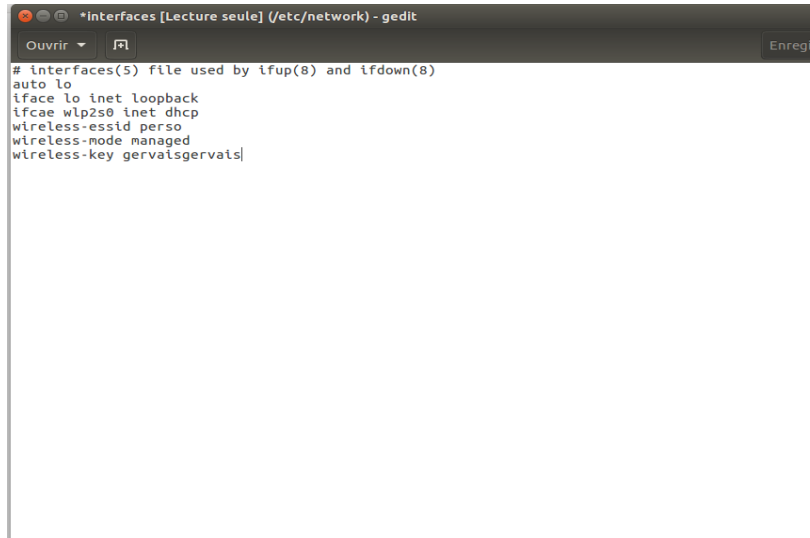
helenrene@HelenRene:~$
```

Illustration 7: Nouvelle adresse IP après changement en ligne de commande

Nous notons sur la figure7 que l'adresse IP de l'interface wlan1 a bien été fixée à « 192.168.1.114 »

-Modification du fichier /etc/network/interfaces

Il nous faut ajouter les lignes présentées à la figure 8 au fichier interfaces en respectant bien la syntaxe. Après modification des configurations, il faut redémarrer le service en exécutant la commande « /etc/init.d/networking restart » afin que la machine puisse prendre en compte la nouvelle configuration.



```
# Interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
iface wlp2s0 inet dhcp
wireless-essid perso
wireless-mode managed
wireless-key gervaisgervais|
```

Illustration 8: Fichier de configuration ouvert et modifié avec gedit

Deuxième partie : Outils de la capture des trames

II. Analyse des protocoles à l'aide des outils pour la capture des trames

➤ Analyse du protocole de résolution d'adresse ARP

Consultons le cache ARP de notre machine, en tapant la commande : « arp -a » pour déterminer les adresses IP qui s'y trouvent. Et nous avons le résultat suivant :

```

junior@junior-HP-15-Notebook-PC: ~
junior@junior-HP-15-Notebook-PC:~$ arp -a
logout.lan (10.223.220.1) à ac:86:74:49:c9:8a [ether] sur wlp2s0f0
junior@junior-HP-15-Notebook-PC:~$

```

Illustration 9 : Contenu du cache arp de notre machine

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 10.223.220.10 » et lançons wireshark pour capturer les trames apr et les trames capturées par cet outil « wireshark » pendant l'exécution des pings sont présentées sur la figure ci-dessous.

The screenshot shows the Wireshark interface with the following details for the selected packet (No. 1):

- Ethernet II**, Src: 10.223.222.5 (9c:d2:1e:c5:f5:e3), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 - Destination: Broadcast (ff:ff:ff:ff:ff:ff)
 - Address: Broadcast (ff:ff:ff:ff:ff:ff)
 - ...1... = LG bit: Locally administered address (this is NOT the factory default)
 - ...1... = IG bit: Group address (multicast/broadcast)
 - Source: 10.223.222.5 (9c:d2:1e:c5:f5:e3)
 - Address: 10.223.222.5 (9c:d2:1e:c5:f5:e3)
 - ...0... = LG bit: Globally unique address (factory default)
 - ...0... = IG bit: Individual address (unicast)
- Type: ARP (0x0806)
- Address Resolution Protocol (request)
 - Hardware type: Ethernet (1)
 - Protocol type: IPv4 (0x0800)
 - Hardware size: 6
 - Protocol size: 4
 - Opcode: request (1)
 - Sender MAC address: 10.223.222.5 (9c:d2:1e:c5:f5:e3)
 - Sender IP address: junior-HP-15-Notebook-PC.lan (10.223.222.5)
 - Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
 - Target IP address: 10.223.220.10 (10.223.220.10)

The packet bytes section shows the raw data in hexadecimal and ASCII format.

Illustration 10 : trames arp capturées lors d'un ping

Nous constatons sur cette capture que la source ayant comme adresse MAC (9c:d2:1e:c5:f5:e3) envoie un message Broadcast à toutes les machines du réseau pour retrouver adresse MAC de la destination ayant comme IP «10.244.225.113» mais seule la destination qui envoie un request de son adresse MAC à la source de la requête comme présenté sur la figure ci-dessous.

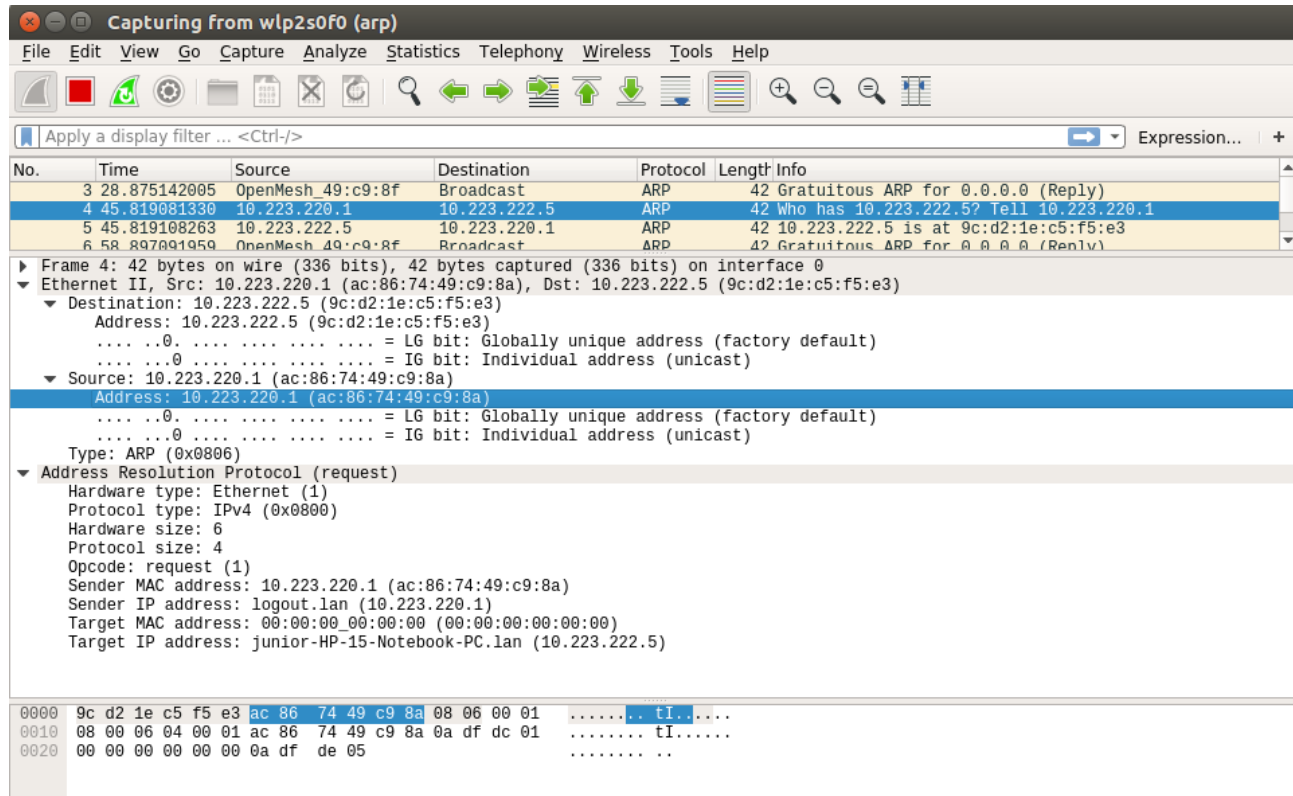


Illustration 11 : trames arp capturées lors d'un request

Une fois que la réponse reçu et ben la source peut peut communiquer avec la destination comme sur la capture suivante.

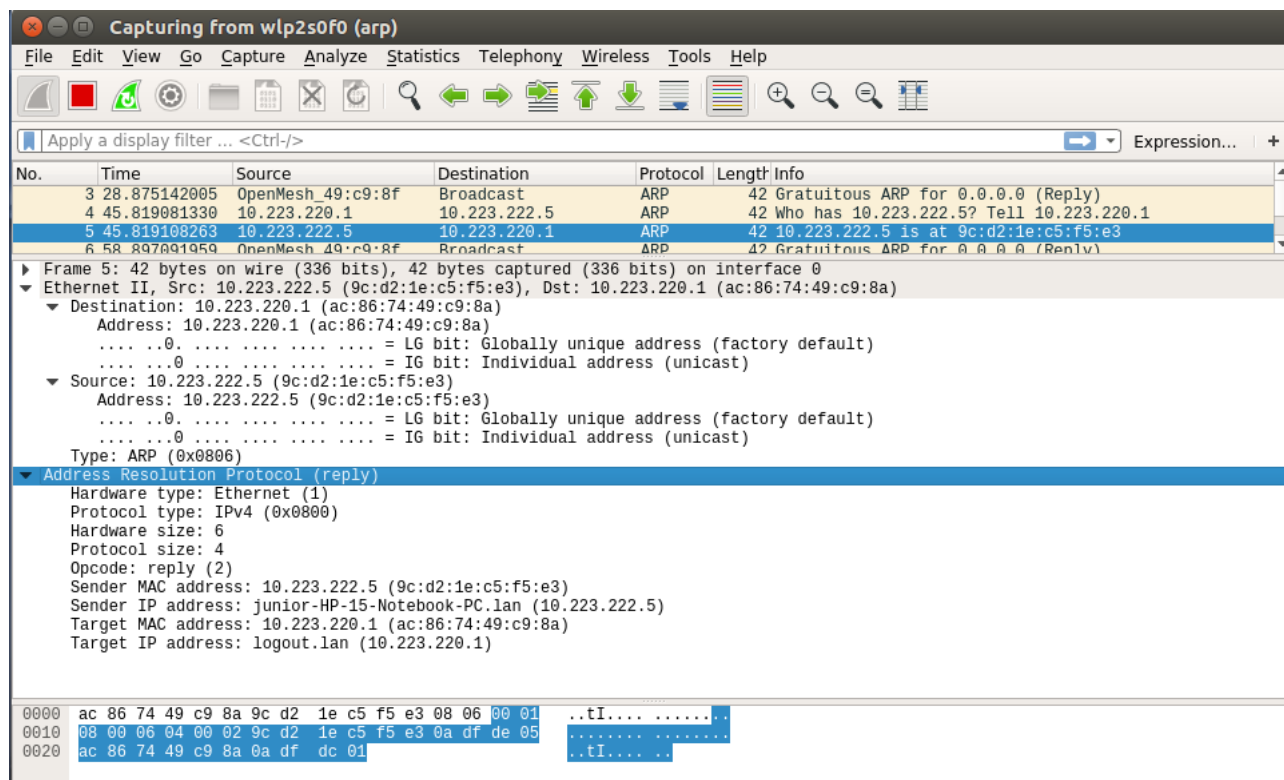


Illustration 12 : trames arp capturées lors d'une communication entre la source et la destination

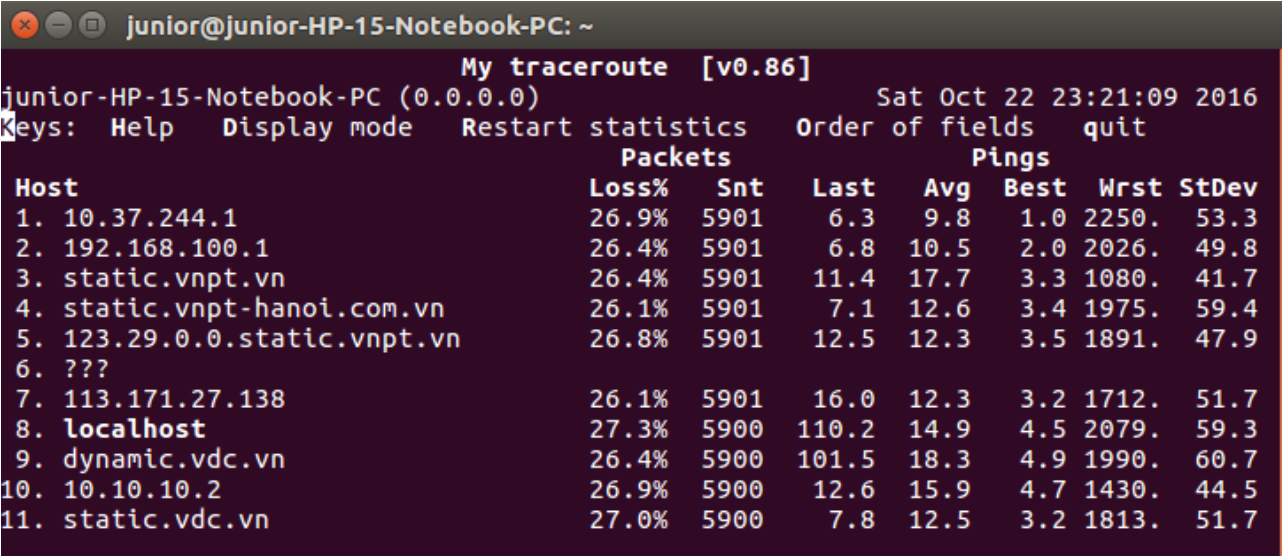
Analyse des routes suivies par les paquets (l'outil mtr)

L'outil *mtr* est une sorte de *traceroute* combiné avec *ping*. C'est un outil de surveillance réseau incomparable. Comme *traceroute*, *mtr* vous indique chaque bond effectué par les paquets pour arriver à destination et comme *ping*, il vous donne pour chaque bond le nombre de paquets perdus, la latence et des données statistiques. Chaque bond (*network hop*) est numéroté comme dans *traceroute* et pour chaque bond, vous est donnée le pourcentage de paquets perdus (*Loss%*), le nombre de paquets envoyés (*Snt*), la latence du dernier paquet envoyé (*Last*) ainsi que la valeur moyenne, la meilleure et la pire (*Avg*, *Best*, *Wrst*). La dernière colonne donne la déviation standard (*StDev*).

mtr permet de diagnostiquer des problèmes de routeurs mal configurés, de firewalls qui bloquent les paquets ICMP. On peut voir dans quel tronçon du réseau, la latence est la plus importante et où on perd le plus de paquets. Tout de même, comme *ping* et *traceroute*, *mtr* utilise des paquets ICMP qui peuvent se retrouver filtrer par des routeurs intermédiaires.

Pour son utilisation, ce n'est pas très compliqué, la prise en main est facile, il suffit de lancer la commande avec une IP ou un domaine. Par exemple *mtr 62.46.251.152* ou *mtr google.com* et *mtr* vous affiche alors en continu les résultats de ses envois de paquets avec statistiques détaillées. Il faut utiliser la touche **q** pour l'arrêter, la touche **d** pour faire un affichage temporel des résultats avec un aspect beaucoup plus visuel, la touche **o** pour changer les colonnes affichées et **h** pour les détails des raccourcis.

A présent, lançons le logiciel pour tracer la route vers le site *www.vnpt.com.vn* avec la commande *mtr www.vnpt.com.vn* et nous obtenons le résultat suivant sur le terminal :



```

junior-HP-15-Notebook-PC (0.0.0.0)
Keys: Help  Display mode  Restart statistics  Order of fields  quit
My traceroute  [v0.86]
Sat Oct 22 23:21:09 2016
Host      Loss%  Snt    Last   Avg    Best   Wrst   StDev
1. 10.37.244.1      26.9%  5901    6.3    9.8    1.0   2250.   53.3
2. 192.168.100.1    26.4%  5901    6.8   10.5    2.0   2026.   49.8
3. static.vnpt.vn   26.4%  5901   11.4   17.7    3.3   1080.   41.7
4. static.vnpt-hanoi.com.vn 26.1%  5901    7.1   12.6    3.4   1975.   59.4
5. 123.29.0.0.static.vnpt.vn 26.8%  5901   12.5   12.3    3.5   1891.   47.9
6. ???
7. 113.171.27.138   26.1%  5901   16.0   12.3    3.2   1712.   51.7
8. localhost        27.3%  5900  110.2   14.9    4.5   2079.   59.3
9. dynamic.vdc.vn   26.4%  5900  101.5   18.3    4.9   1990.   60.7
10. 10.10.10.2       26.9%  5900   12.6   15.9    4.7   1430.   44.5
11. static.vdc.vn    27.0%  5900    7.8   12.5    3.2   1813.   51.7

```

Illustration 13 : Route de la machine source vers le site *www.vnpt.com.vn* en utilisant *mtr*

Nous remarquons que sur notre terminal, nous avons l’affichage qui nous donne pour chaque bond (host) le nombre de paquets perdus, la latence et des données statistiques.

Et en lançant wireshark pour voir et capturer l’envoi et réception des trames. Et en se basant au niveau IP, recherchons les champs qui varient entre les envois successifs de paquets (excepté le champ identification et checksum qui ne sont d’aucun intérêt).

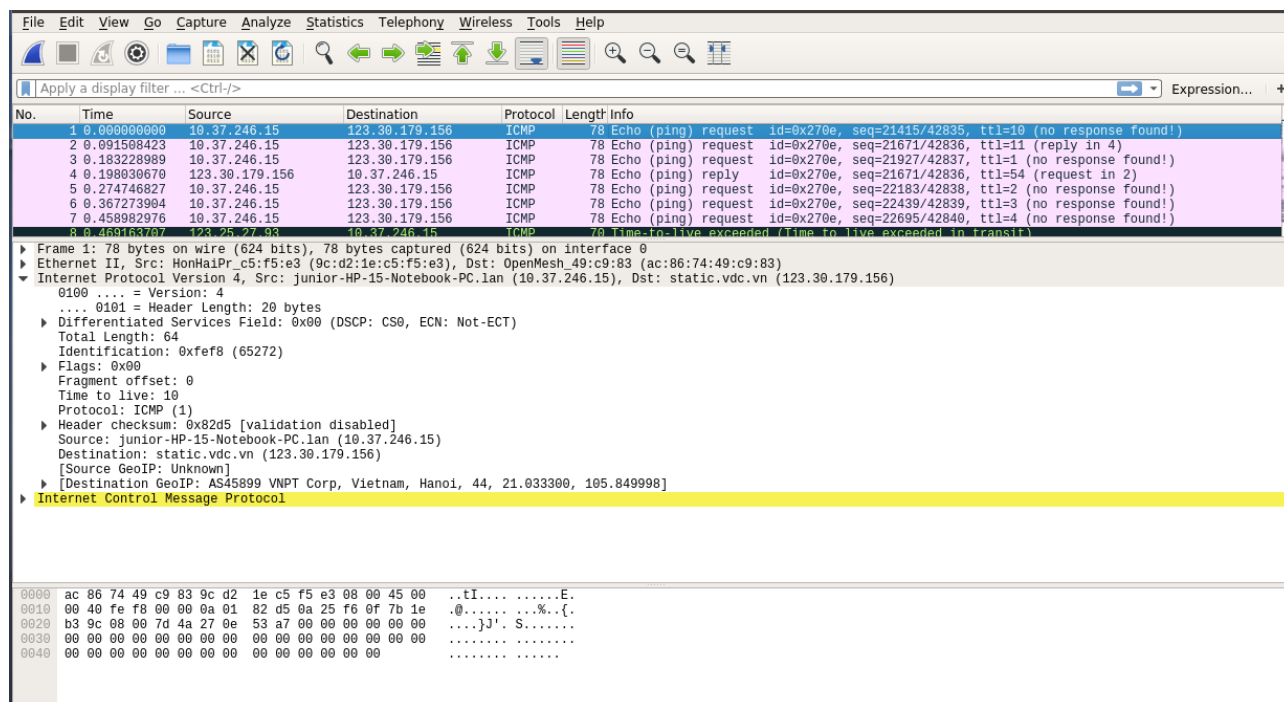


Illustration 14 : Trames de route de la machine source vers le site www.vnpt.com.vn en utilisant mtr

La succession des adresses IP affichées par rapport à l'ensemble des interfaces réseaux réellement traversées pour joindre la destination, nous remarquons que pour atteindre la destination, le paquet quitte de l'adresse de mon ordinateur (adresse source) en passant par plusieurs hosts intermédiaires pour atteindre la destination et en plus sur la capture, il y a pour un envoi des paquets entre la source et le host, un nombre de paquet envoyé, celui perdu entre les deux entités jusqu'à la destination.

La topologie du réseau ici est en étoile, vu que les paquets traverse plusieurs bonds pour atteindre la destination, et plus particulièrement le plan d'adressage est tel que il y a présence de 7 réseaux différents entre la source et la destination comme le montre notre capture qui va du réseau 10.37.246.15 jusqu'à la succession des interfaces traversées par nos paquets de 123.30.179.156 vers 123.25.27.93, puis 123.29.1.189 puis 113.171.33.222 puis 10.37.244.1 puis 192.168.100.1 puis 14.177.160.1 puis la destination,

En nous basant exclusivement sur la capture de trames, ce logiciel fonctionne tel qu'en exécutant la commande «mtr www.vnpt.com.vn» sur le terminal, ce logiciel nous donne un nombre de bon (*network hop*) numéroté, chaque bond (*network hop*) effectué par les paquets pour arriver à destination, et pour chaque bond (*network hop*), nous avons le nombre de paquets perdus, la latence et des données statistiques, en plus nous avons aussi le pourcentage de paquets perdus (*Loss%*), le nombre de paquets envoyés (*Snt*), la latence du dernier paquet envoyé (*Last*) ainsi que la valeur moyenne, la meilleure et la pire (*Avg*, *Best*, *Wrst*). La dernière colonne donne la déviation standard (*StDev*).

Analyse du protocole TCP

Pour lancer la capture avec tcpdump dans un fichier, nous devons taper la commande : `sudo tcpdump port http -w capture.tcpdump`

En lançant la commande :

`wget http://fad.ifi.edu.vn/itifad/file.php/28/documents/WS_user-guide-a4.pdf`. On lance la capture avec la commande `sudo tcpdump -v -i wlp2s0f0 'port 80'` et en arrêtant la capture après avoir lancé le téléchargement on peut analyser les résultats suivants:

Phase de connexion

Sur tcpdump on a :

junior-HP-15-Notebook-PC.lan.36530 > 112.137.140.42.http: Flags [P.], cksum 0x4084 (correct), seq 1:191, ack 1, win 229, options [nop,nop,TS val 2289983 ecr 665095620], length 190: HTTP, length: 190

GET /itifad/file.php/28/documents/WS_user-guide-a4.pdf HTTP/1.1

User-Agent: Wget/1.17.1 (linux-gnu)

Accept: */*

Accept-Encoding: identity

Host: fad.ifi.edu.vn

Connection: Keep-Alive

Sur wireshark on aura :

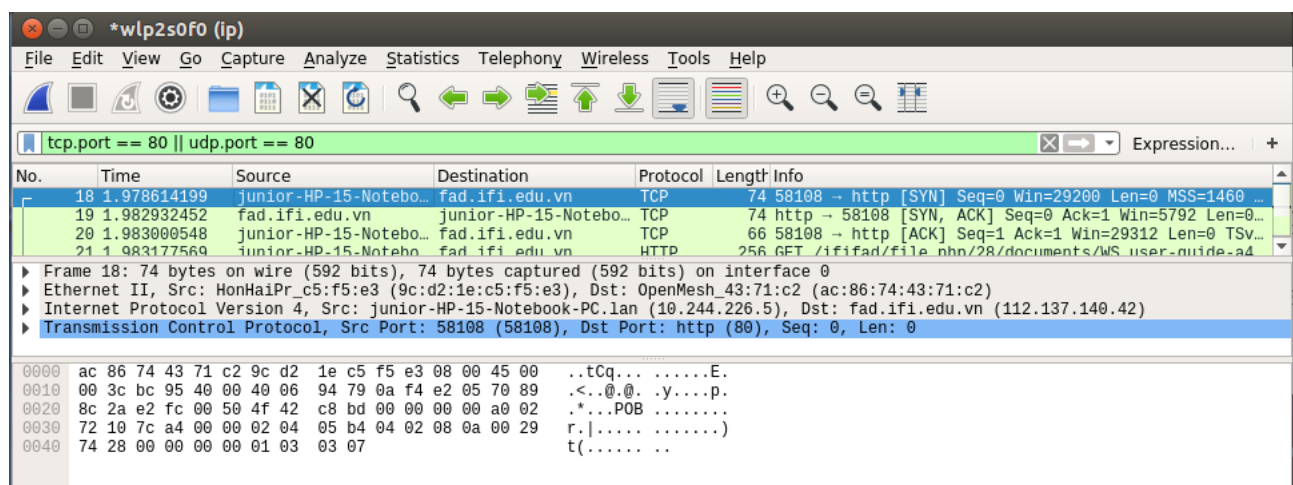


Illustration 15 : Phase de connexion entre deux machines avant le téléchargement

Phase de transfert de données

Sur tcpdump on a :

112.137.140.42.http > junior-HP-15-Notebook-PC.lan.36530: Flags [P.], cksum 0x3a45 (correct), seq 1:757, ack 191, win 54, options [nop,nop,TS val 665095702 ecr 2289983], length 756: HTTP, length: 756

HTTP/1.1 303 See Other

Date: Sun, 23 Oct 2016 06:27:04 GMT

Server: Apache/2.2.16 (Debian)

X-Powered-By: PHP/5.3.3-7+squeeze19

Set-Cookie: MoodleSession=o67bublkau7kiu0gi2qd1tlai2; path=/

Expires: Thu, 19 Nov 1981 08:52:00 GMT

Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0

Pragma: no-cache

Set-Cookie: MoodleSessionTest=xLQYiaTq1y; path=/

Location: http://fad.ifi.edu.vn/ififad/login/index.php

Content-Length: 216

Keep-Alive: timeout=15, max=100

Connection: Keep-Alive

Content-Type: text/html

```
<meta http-equiv="refresh" content="0; url=http://fad.ifi.edu.vn/ififad/login/index.php"
/><script type="text/javascript">
  //<![CDATA[
    location.replace('http://fad.ifi.edu.vn/ififad/login/index.php');
  //]]>
</script>[!http]
```


Sur wireshark on aura :

Wireshark packet capture showing a series of HTTP and TCP packets between junior-HP-15-Notebook-PC.lan and fad.ifi.edu.vn. The capture is filtered for tcp.port == 80 || udp.port == 80. The selected packet is a TCP ACK segment (Seq=757, Ack=434, Win=7936) from the notebook to the server. Below the packet list, the packet details pane shows the frame structure: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol.

No.	Time	Source	Destination	Protocol	Length	Info
18	1.978614199	junior-HP-15-Notebo...	fad.ifi.edu.vn	TCP	74	58108 → http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 ...
19	1.982932452	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	74	http → 58108 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0...
20	1.983000548	junior-HP-15-Notebo...	fad.ifi.edu.vn	TCP	66	58108 → http [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSv...
21	1.983177569	junior-HP-15-Notebo...	fad.ifi.edu.vn	HTTP	256	GET /ififad/file.php/28/documents/WS_user-guide-a4...
22	1.987731432	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	66	http → 58108 [ACK] Seq=1 Ack=191 Win=6912 Len=0 TS...
27	2.660109881	fad.ifi.edu.vn	junior-HP-15-Notebo...	HTTP	822	HTTP/1.1 303 See Other (text/html)
28	2.660155069	junior-HP-15-Notebo...	fad.ifi.edu.vn	TCP	66	58108 → http [ACK] Seq=191 Ack=757 Win=30720 Len=0...
30	2.661390757	junior-HP-15-Notebo...	fad.ifi.edu.vn	HTTP	309	GET /ififad/login/index.php HTTP/1.1
31	2.667504402	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	66	http → 58108 [ACK] Seq=757 Ack=434 Win=7936 Len=0 ...
33	3.957531866	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	1466	[TCP segment of a reassembled PDU]
34	3.958655455	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	1466	[TCP segment of a reassembled PDU]
35	3.958696809	junior-HP-15-Notebo...	fad.ifi.edu.vn	TCP	66	58108 → http [ACK] Seq=434 Ack=3557 Win=36352 Len=...
36	3.959607989	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	1466	[TCP segment of a reassembled PDU]
37	3.960016081	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	1466	[TCP Out-Of-Order] http → 58108 [ACK] Seq=757 Ack=...
38	3.960037819	junior-HP-15-Notebo...	fad.ifi.edu.vn	TCP	78	58108 → http [ACK] Seq=434 Ack=4957 Win=39168 Len=...
39	3.969147608	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	1466	[TCP segment of a reassembled PDU]
40	3.969184242	junior-HP-15-Notebo...	fad.ifi.edu.vn	TCP	66	58108 → http [ACK] Seq=434 Ack=6357 Win=41984 Len=...
41	3.972578527	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	1466	[TCP segment of a reassembled PDU]
42	3.972617719	junior-HP-15-Notebo...	fad.ifi.edu.vn	TCP	66	58108 → http [ACK] Seq=434 Ack=7757 Win=44800 Len=...
43	3.975667845	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	1466	[TCP segment of a reassembled PDU]
44	3.975738207	junior-HP-15-Notebo...	fad.ifi.edu.vn	TCP	66	58108 → http [ACK] Seq=434 Ack=9157 Win=47616 Len=...
45	3.977482815	fad.ifi.edu.vn	junior-HP-15-Notebo...	TCP	1466	[TCP segment of a reassembled PDU]
46	3.977510901	junior-HP-15-Notebo...	fad.ifi.edu.vn	TCP	66	58108 → http [ACK] Seq=434 Ack=10557 Win=50432 Len=...
47	3.979187885	fad.ifi.edu.vn	junior-HP-15-Notebo...	HTTP	1266	HTTP/1.1 200 OK (text/html)

▶ Frame 18: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 ▶ Ethernet II, Src: HonHaiPr_c5:f5:e3 (9c:d2:1e:c5:f5:e3), Dst: OpenMesh_43:71:c2 (ac:86:74:43:71:c2)
 ▶ Internet Protocol Version 4, Src: junior-HP-15-Notebook-PC.lan (10.244.226.5), Dst: fad.ifi.edu.vn (112.137.140.42)
 ▶ Transmission Control Protocol, Src Port: 58108 (58108), Dst Port: http (80), Seq: 0, Len: 0

```

0000  ac 86 74 43 71 c2 9c d2 1e c5 f5 e3 08 00 45 00  ..tCq... ..E.
0010  00 3c bc 95 40 00 40 06 94 79 0a f4 e2 05 70 89  <..@. .y....p.
0020  8c 2a e2 fc 00 50 4f 42 c8 bd 00 00 00 00 a0 02  .*...POB .....
0030  72 10 7c a4 00 00 02 04 05 b4 04 02 08 0a 00 29  r.|.....)
0040  74 28 00 00 00 00 01 03 03 07                    t(.....
  
```

Illustration 16 : Phase de transfert de données entre deux machines avant le téléchargement

Phase de déconnexion

Sur tcpdump on aura :

junior-HP-15-Notebook-PC.lan.36530 > 112.137.140.42.http: Flags [.] , cksum 0x3994 (correct), ack 11758, win 415, options [nop,nop,TS val 2290457 ecr 665096102], length 0

Sur wireshark on aura :

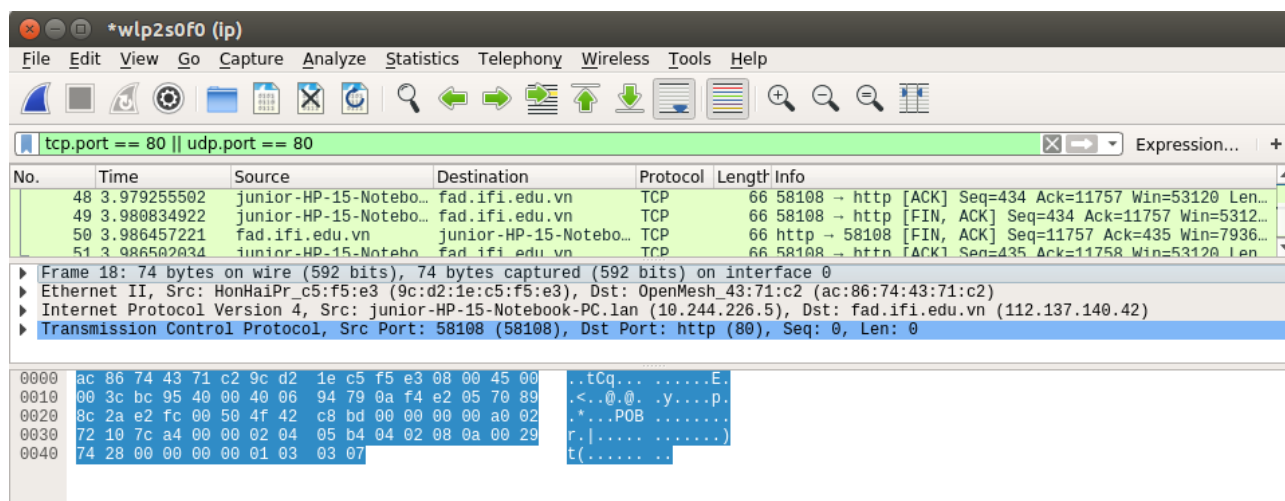


Illustration 17 : Phase de déconnexion de données entre deux machines avant le téléchargement

Nous pouvons identifier des numéros initiaux de séquence (Initial Sequence Number - ISN) utilisés dans les deux sens comme suit :

junior-HP-15-Notebook-PC.lan.36530 > 112.137.140.42.http: Flags [P.], cksum 0x4084 (correct), seq 1:191, ack 1, win 229, options [nop,nop,TS val 2289983 ecr 665095620], length 190: HTTP, length: 190

112.137.140.42.http > junior-HP-15-Notebook-PC.lan.36530: Flags [P.], cksum 0x3a45 (correct), seq 1:757, ack 191, win 54, options [nop,nop,TS val 665095702 ecr 2289983], length 756: HTTP, length: 756

Représentons alors sur un schéma cette échange en montrant l'évolution des numéros de séquence, de la fenêtre, des acquittements, des fanions, etc comme suit :

- ✓ Flags [P.], cksum 0x4084 (correct), seq 1:191, ack 1, win 229
- ✓ Flags [S.], cksum 0x2953 (correct), seq 4035419922, ack 2957028589, win 5792
- ✓ Flags [.], cksum 0x6da9 (correct), ack 1, win 229
- ✓ Flags [P.], cksum 0x4084 (correct), seq 1:191, ack 1, win 229
- ✓ Flags [.], cksum 0x6d99 (correct), ack 191, win 54
- ✓ Flags [P.], cksum 0x3a45 (correct), seq 1:757, ack 191, win 54
- ✓ Flags [.], cksum 0x67ea (correct), ack 434, win 62
- ✓ Flags [.], cksum 0x0fb6 (correct), seq 757:2157, ack 434, win 62
- ✓ Flags [.], cksum 0x3994 (correct), ack 11758, win 415

➤ Analyse du protocole telnet et la capture des informations qui circulent en clair sur le réseau

Analysons ce protocole telnet pour prouver l'existence de la transparence des informations des utilisateurs sur le réseau à connexion non sécurisée, alors connectons ordinateur source au serveur Telnet lancé à partir d'un ordinateur en tapant la commande « telnet 10.244.225.113 » en s'identifiant avec notre nom de compte et password.

On a les informations suivantes :

- le port source 40605 et le port destination est 23.
- le mot de passe en analysant la capture des trames obtenues de Wireshark, on retrouve le mot de passe entré en clair et le login comme le réseau n'est pas sécurisé (pas de chiffrement).

Réseau Ethernet	Réseau sans fil
Les communications sont de type unicast et multicast.	Les communications sont de type broadcast.
On ne capture que les données qui nous sont destinées.	On capture que toutes les données qui sont disponibles ou envoyées sur le réseau peu importe leur destination.
Nous ne visualisons que les données à notre destination.	Nous visualisons toutes les données qui sont disponibles ou envoyées sur le réseau peu importe leur destination.

Conclusion

En définitive, au cours de ce projet, nous avons eu la main mise les commandes unix pour nous permettre une utilisation familière en fournissant des informations clés sur la configuration des interfaces réseaux en utilisant les commandes unix.

L'utilisation du 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 emettrice en Broadcast vers toutes les machines du réseau mais seule la machine dont l'adresse IP correspond à celle contenue dans la requête répond au message en envoyant son adresse MAC. Le cache arp de la machine emettrice est alors mis à jour et la communication peut dès lors se dérouler librement entre les deux machines et à chaque phase avons analysé ledit protocole avec les outils d'analyse réseau tels que tcpdump et wireshark.

Pour arriver à comprendre le protocole TCP, nous avons besoin de comprendre ces trois étapes qui partent de la phase de connexion pendant laquelle le client et le serveur s'échangent des paquets de synchronisation et des ACK, de la phase de transfert des données et enfin de la phase de déconnexion ou de la fin de la connexion entre le client et le serveur. A chaque niveau une analyse de réseau au moyen du logiciel wireshark est appliquée comme vous constaterez sur les captures ci-haut.

Références

Michel Bertrand - Université de Toulouse-Institut Universitaire de France Sandro Guzzi-Heeb - Université de Lausanne, Suisse Claire Lemerrier - Centre de sociologie des organisations (UMR 7116 CNRS-Sciences Po). Introduction : où en est l'analyse de réseaux en histoire?. Vol. 21, #1, Décembre 2011.

<http://www.leunen.com/linux/2011/06/mtr-un-outil-de-diagnostique-reseau/>

<http://crash-blog.com/surveillance-reseau-mtr/>

<http://www.octetmalin.net/linux/tutoriels/tcpdump-ecouter-capturer-paquet-ip-reseau.php>