Architecture des réseaux – TP2 Encadré par PG

Jonathan Morgado-Samagaio Groupe : 3A

Compte-rendu de TP réseau

Wireshark et HTTP

Table des matières

Réponses aux questions :

Exercice 1 : Capture

Nous commençons par récupérer l'adresse IP de notre machine et le nom de l'interface réseau.

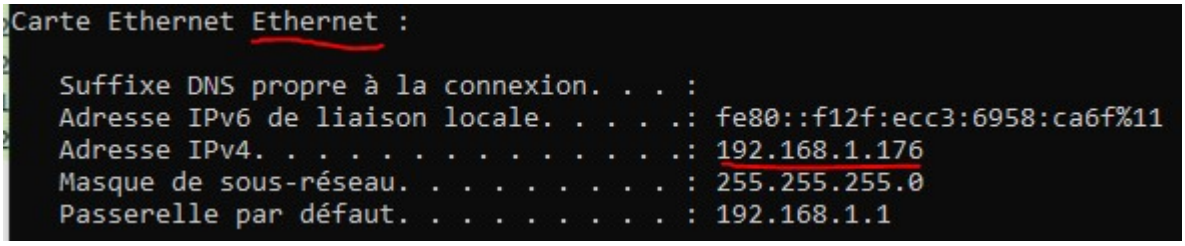


Figure 1 : Résultat de ipconfig /all

Mon adresse est 192.168.1.176 et le nom de l'interface est Ethernet. En ayant la capture Wireshark d'activé, nous ouvrons le site et obtenons :

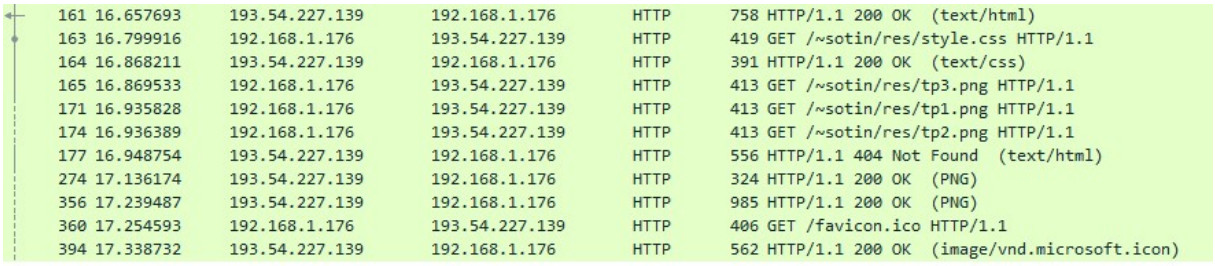


Figure 2 : Résultat de la capture Wireshark

Exercice 2 : Pile réseau

Nous allons examiner les différents éléments de la pile réseau.

```
Transmission Control Protocol, Src Port: 49860, Dst Port: 80, Seq: 1, Ack: 1, Len: 388
```

Figure 3 : Protocole TCP

Pour le protocole TCP, nous avons comme port source le port 49860 et comme port de destination le port 80.

```
Internet Protocol Version 4, Src: 192.168.1.176, Dst: 193.54.227.139
```

Figure 4 : Protocole IP

Pour le protocole IP, l'identifiant source est le : 192.168.1.176 et l'identifiant de destination est le : 193.54.227.139. Ce sont des adresses IP.

Nous savons que la source est mon pc avec le résultat de ipconfig /all du premier exercice.

```
C:\Users\Pack>tracert 193.54.227.139

Détermination de l'itinéraire vers 193.54.227.139 avec un maximum de 30 sauts.

  1     2 ms     1 ms     1 ms  homerouter.cpe [192.168.1.1]
  2     *        *        *     Délai d'attente de la demande dépassé.
  3     *        *        *     Délai d'attente de la demande dépassé.
  4    40 ms    38 ms    39 ms  192.168.255.2
  5    39 ms    53 ms    39 ms  192.168.255.3
  6    48 ms    61 ms    43 ms  194.149.185.10
  7    55 ms     *        *     th2-6k-3.routers.proxad.net [212.27.40.121]
  8    54 ms    29 ms    37 ms  193.51.187.208
  9    67 ms    69 ms    55 ms  te-0-1-0-15-ren-nr-lyon2-rtr-091.noc.renater.fr [193.51.177.8]
 10    74 ms    57 ms    52 ms  te2-5-marseille2-rtr-021.noc.renater.fr [193.51.177.196]
 11    58 ms    61 ms    53 ms  193.51.180.119
 12    61 ms    70 ms    73 ms  te-0-1-0-15-ren-nr-montpellier-rtr-091.noc.renater.fr [193.51.180.192]
 13    56 ms    58 ms    58 ms  te0-0-0-15-ren-nr-toulouse-rtr-091.noc.renater.fr [193.55.203.224]
 14    80 ms    77 ms    58 ms  remip-2000-te1-3-toulouse-rtr-021.noc.renater.fr [193.51.181.177]
 15     *        *        *     Délai d'attente de la demande dépassé.
 16     *        *        *     Délai d'attente de la demande dépassé.
 17     *        *        *     Délai d'attente de la demande dépassé.
 18     *        *        *     Délai d'attente de la demande dépassé.
 19     *        *        *     Délai d'attente de la demande dépassé.
 20     *        *        *     Délai d'attente de la demande dépassé.
 21     *        *        *     Délai d'attente de la demande dépassé.
 22     *        *        *     Délai d'attente de la demande dépassé.
 23     *        *        *     Délai d'attente de la demande dépassé.
 24     *        *        *     Délai d'attente de la demande dépassé.
 25     *        *        *     Délai d'attente de la demande dépassé.
```

Figure 5 : Résultat tracert 193.54.227.139

Nous pouvons voir que la source se trouve à Toulouse. Ca doit être le serveur de l'IUT que nous ne pouvons pas voir à cause des sécurités.

```
Ethernet II, Src: HonHaiPr_86:15:bb (d0:27:88:86:15:bb), Dst: HuaweiTe_72:7d:5f (04:79:70:72:7d:5f)
```

Figure 6 : Couche Ethernet

Pour le protocole Ethernet, l'identifiant source est le : d0:27:88 :86:15:bb et l'identifiant de destination est le : 04:79:70:72:7d:5f. Ce sont des adresses MAC.

```
C:\Users\Pack>arp -a

Interface : 192.168.1.176 --- 0xb
  Adresse Internet      Adresse physique      Type
  192.168.1.1           04-79-70-72-7d-5f     dynamique
  192.168.1.174         50-d4-f7-f5-9a-a0     dynamique
  192.168.1.255         ff-ff-ff-ff-ff-ff     statique
  224.0.0.2             01-00-5e-00-00-02     statique
  224.0.0.22            01-00-5e-00-00-16     statique
  224.0.0.251           01-00-5e-00-00-fb     statique
  224.0.0.252           01-00-5e-00-00-fc     statique
  239.255.255.250       01-00-5e-7f-ff-fa     statique
  255.255.255.255       ff-ff-ff-ff-ff-ff     statique
```

Figure 7 : Résultat de arp –a

Nous pouvons voir que l'adresse de destination est l'adresse de mon routeur.

L'adresse source doit être l'adresse du serveur de l'IUT.

L'encapsulation consiste a mettre les données d'un protocole dans un autre.

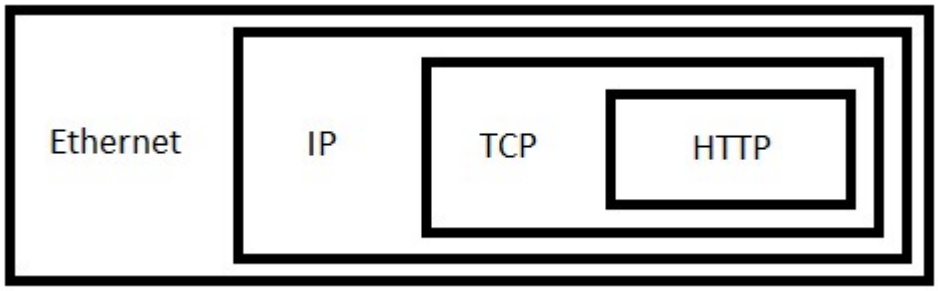


Figure 8 : Croquis de l'encapsulation

274	17.136174	193.54.227.139	192.168.1.176	HTTP	324 HTTP/1.1 200 OK (PNG)

Figure 9 : Message PNG sélectionné

```
[33 Reassembled TCP Segments (44558 bytes): #181(1384), #182(1384), #184(1384), #185(1384), #187(1384), #188(1384), #190(1384), #191(1384), #193(1384), #194(1384), #212(1384), #213(1384),
```

Figure 10 : Fragments du message

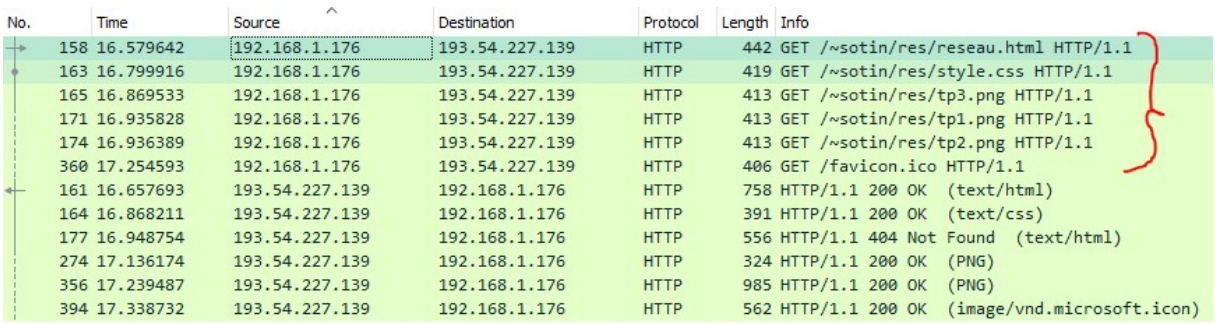
Nous pouvons voir qu'il y a 33 fragments. La taille maximale est de 1384 octets. La taille totale du message st de 44 558 octets et c'est une image.

Exercice 3 : Dialogue http

Voici le lien : http://choregies.iut-blagnac.fr/~sotin/res/reseau.html

Pour afficher ce lien dans une page HTML, nous utilisons la balise <a>.

Dans cette URL, le champ host est : choregies.iut-blagnac.fr, le port n'étant pas visible, celui par défaut est utilisé: c'est le port 80. Le champ chem_abs est : /~sotin/res/reseau.html



No.	Time	Source	Destination	Protocol	Length	Info
158	16.579642	192.168.1.176	193.54.227.139	HTTP	442	GET /~sotin/res/reseau.html HTTP/1.1
163	16.799916	192.168.1.176	193.54.227.139	HTTP	419	GET /~sotin/res/style.css HTTP/1.1
165	16.869533	192.168.1.176	193.54.227.139	HTTP	413	GET /~sotin/res/tp3.png HTTP/1.1
171	16.935828	192.168.1.176	193.54.227.139	HTTP	413	GET /~sotin/res/tp1.png HTTP/1.1
174	16.936389	192.168.1.176	193.54.227.139	HTTP	413	GET /~sotin/res/tp2.png HTTP/1.1
360	17.254593	192.168.1.176	193.54.227.139	HTTP	406	GET /favicon.ico HTTP/1.1
161	16.657693	193.54.227.139	192.168.1.176	HTTP	758	HTTP/1.1 200 OK (text/html)
164	16.868211	193.54.227.139	192.168.1.176	HTTP	391	HTTP/1.1 200 OK (text/css)
177	16.948754	193.54.227.139	192.168.1.176	HTTP	556	HTTP/1.1 404 Not Found (text/html)
274	17.136174	193.54.227.139	192.168.1.176	HTTP	324	HTTP/1.1 200 OK (PNG)
356	17.239487	193.54.227.139	192.168.1.176	HTTP	985	HTTP/1.1 200 OK (PNG)
394	17.338732	193.54.227.139	192.168.1.176	HTTP	562	HTTP/1.1 200 OK (image/vnd.microsoft.icon)

Figure 11 : Résultat de Wireshark trié par sources

L'unique méthode http utilisé par mon navigateur est la méthode GET. Le protocole utilisé est le 1.1.

Format des réponses : Ligne_état = Version_HTTP SP Code_état SP Raison CRLF

Dans les colonnes requête et réponse sont données les numéros des lignes Wireshark.

Requête	Réponse	Commentaire
158	161	Demande du corps de la page par le client et réponse positive du serveur.
163	164	Demande de la feuille de style par le client et réponse positive du serveur.
165	177	Demande d'une image par le client et réponse négative du serveur.
171	274	Demande d'une image par le client et réponse positive du serveur.
174	356	Demande d'une image par le client et réponse négative du serveur.
360	394	Demande de l'icône par le client et réponse positive du serveur.

On nous a demandé de vider le cache avant la capture pour que toutes les requêtes soit effectué et pas que l'on récupère une donnée dans le cache.

Ce qui déclenche la première requête est l'appuie sur le lien qui demande au serveur le corps de la page. Un fois le corps de la page reçu, le client demande chaque information qu'il ne possède pas, donc ici la feuille de style, les images et l'icône.

Conclusion :

Dans ce TP, nous avons vu comment utiliser Wireshark pour voir toutes les requêtes et réponses effectuées lors du chargement d'une page. Nous avons ensuite vu comment se découpais une

requête avec les différents éléments de la pile comme la couche TCP, la couche IP et la couche Ethernet. Nous avons aussi vu que des messages lourds peuvent être fragmentés pour respecter la limite de poids. Nous avons ensuite vu comment se découper une URL et les messages reçu par Wireshark et pourquoi il y a eu toutes ces requêtes.