

TP 2 Couche Transport

Table des matières



Objectifs	3
I - TP Réseaux Couche Transport	4
1. Three way handshake et flags	4
II - UDP broadcast (Exercice Optionnel +2 dans la note TP finale du semestre)	
7	

Objectifs

- Analyser l'entête TCP avec Wireshark ;
- Travail à rendre : pour chaque question/étape prenez une capture d'écran *complète* (les captures d'écran partielles ne seront comptabilisés).

TP Réseaux Couche Transport

1. Three way handshake et flags

La réalisation de ce TP nécessite l'installation de Wireshark :

<https://www.wireshark.org/download.html>

Lors du lancement de Wireshark soyez sûr que le logiciel est lancé en mode administrateur .

Ouvrez Wireshark et sélectionnez l'interface Npcap Loopback (1). Ensuite, insérez le filtre de port [`tcp.port == 9500`] en haut (2) afin d'observer seulement les paquets destinés / générés par le port TCP 9500.

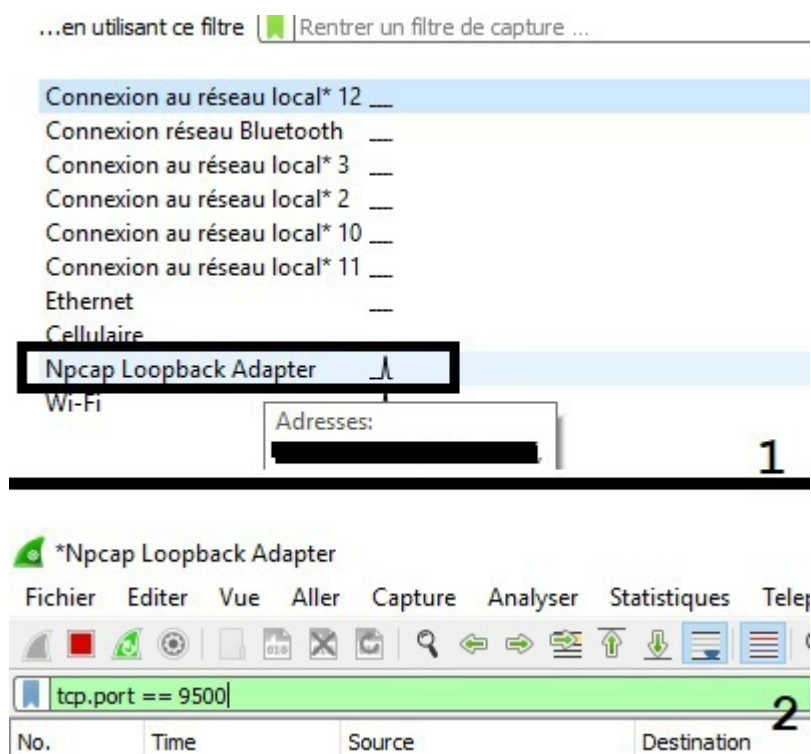


Figure 1. Interface loopback

Ci-dessous les boutons permettant de a) d'arrêter la capture, b) redémarrer la capture, c) changer l'interface de capture.

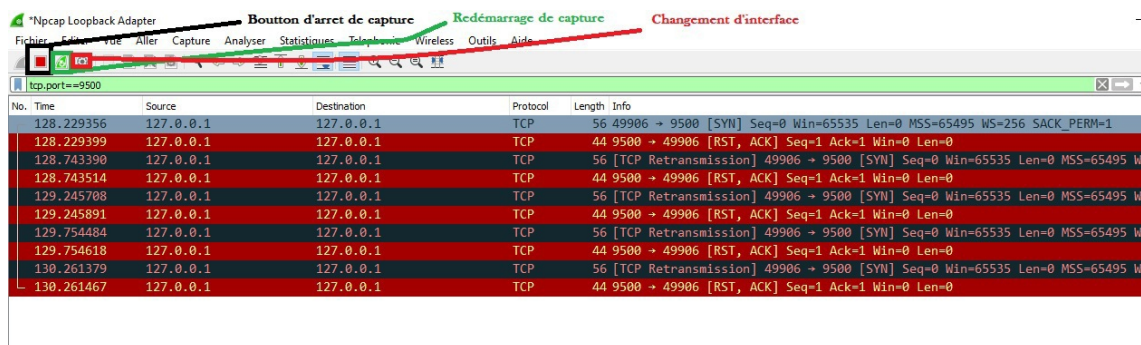


Figure 2. Interface de capture

Question

1. Exécutez le client (Client.py) sans lancer le serveur et observez le résultat sur Wireshark. Quels sont les flags affichés dans la capture Wireshark ?
2. Exécutez le serveur (Serveur.py du TP 1) et ensuite le programme Client et observez le résultat sur Wireshark. Repérez les flags utilisés pour l'échange three-way-handshake dans la capture (les trois premiers segments échangés entre le client et serveur).
3. Transmettez un message du client vers le serveur et repérez le contenu de la transmission sur Wireshark (voir Figure 3).
4. Quels sont les Flags utilisés dans le segment contenant les données envoyé par client ?
5. Quels sont les options négociés initialement entre le serveur et client (sélectionnez les deux premiers échanges du three-way handshake afin de visualiser ces options)?
6. Transmettez un message "Fin" depuis le client vers le serveur afin de clôturer la connexion entre serveur et client. Repérez les flags utilisés pour clôturer la connexion.

8.602153	127.0.0.1	127.0.0.1	TCP	56	50059 → 9500 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
8.602233	127.0.0.1	127.0.0.1	TCP	56	9500 → 50059 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
8.602296	127.0.0.1	127.0.0.1	TCP	44	50059 → 9500 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
27.415797	127.0.0.1	127.0.0.1	TCP	66	50059 → 9500 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=22
27.415958	127.0.0.1	127.0.0.1	TCP	44	9500 → 50059 [ACK] Seq=1 Ack=23 Win=2619648 Len=0

```

Frame 148: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
Null/Loopback
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
Transmission Control Protocol, Src Port: 50059, Dst Port: 9500, Seq: 1, Ack: 1, Len: 22
Data (22 bytes)

```

```

0000 02 00 00 00 45 00 00 3e 8c de 40 00 80 06 00 00 .....E...>..@.....
0010 7f 00 00 01 7f 00 00 01 c3 8b 25 1c 64 4f dc 3c .....<.....
0020 5a e3 1b 21 50 18 27 f9 ad 81 00 00 4d 45 53 53 Z..!P.. ...MESS
0030 41 47 45 20 56 45 52 53 20 70 6f 72 74 20 39 35 AGE VERS port 95
0040 30 30

```

Figure 3. Capture du contenu du message

UDP broadcast (Exercice Optionnel +2 dans la note TP finale du semestre)

Afin d'obtenir le +2pts, vous devez rendre une capture d'écran complète de votre écran d'un broadcast UDP sur le port 5000 (si l'image est altérée, réduite ou coupée celle-ci ne sera pas compté)

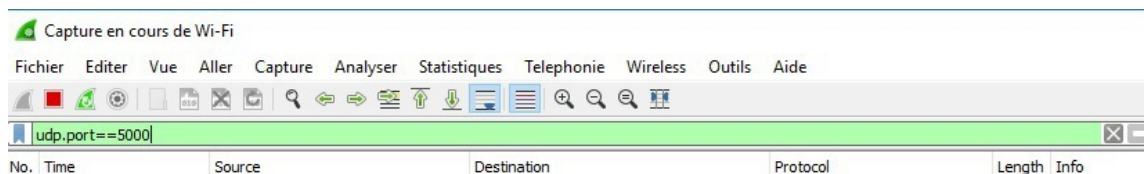


Figure 4. Filtre UDP

1. Connectez au réseau local.
2. Lancez Wireshark sur la machine de votre camarade ou d'une machine quelconque dans le réseau local (votre camarade doit être connecté au réseau local aussi).
3. Filtrez pour obtenir les messages UDP entrants du port 5000 sur la machine de votre camarade (`udp.port==5000`);
4. Exécutez le programme suivant sur votre machine afin de diffusez un message dans le réseau (le code est téléchargeable depuis le lien suivant <https://drive.google.com/file/d/1jhZyTtLtDm6RARtjh2S5IPMy0xGxaT9I/view?usp=sharing>) :

```

1 import socket
2 SocketUDP= socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # ceration socket UDP
3 SocketUDP.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1) # rendre le
   socket UDP broadcast
4 Numero_Port=5000
5 AdresseDiffusion='192.168.1.255'
6 MessageADiffuser='Votre message a diffuser dans le reseau' # contenu du message
   diffusion
7 SocketUDP.sendto(MessageADiffuser, (AdresseDiffusion, Numero_Port)) #
   transmettre le message
8

```