1 LE PROTOCOLE TCP ET UDP

1.1. Exercice 1 - TCP

Sur votre VM Kali Linux, ouvrez deux **shells**.

1. Exécutez **netcat** en mode serveur TCP sur le port 5000.

   `nc -l -p 5000`

2. Réalisez une capture des échanges entre le serveur et le client et décoder le niveau TCP en utilisant wireshark sur l'interface loopback.

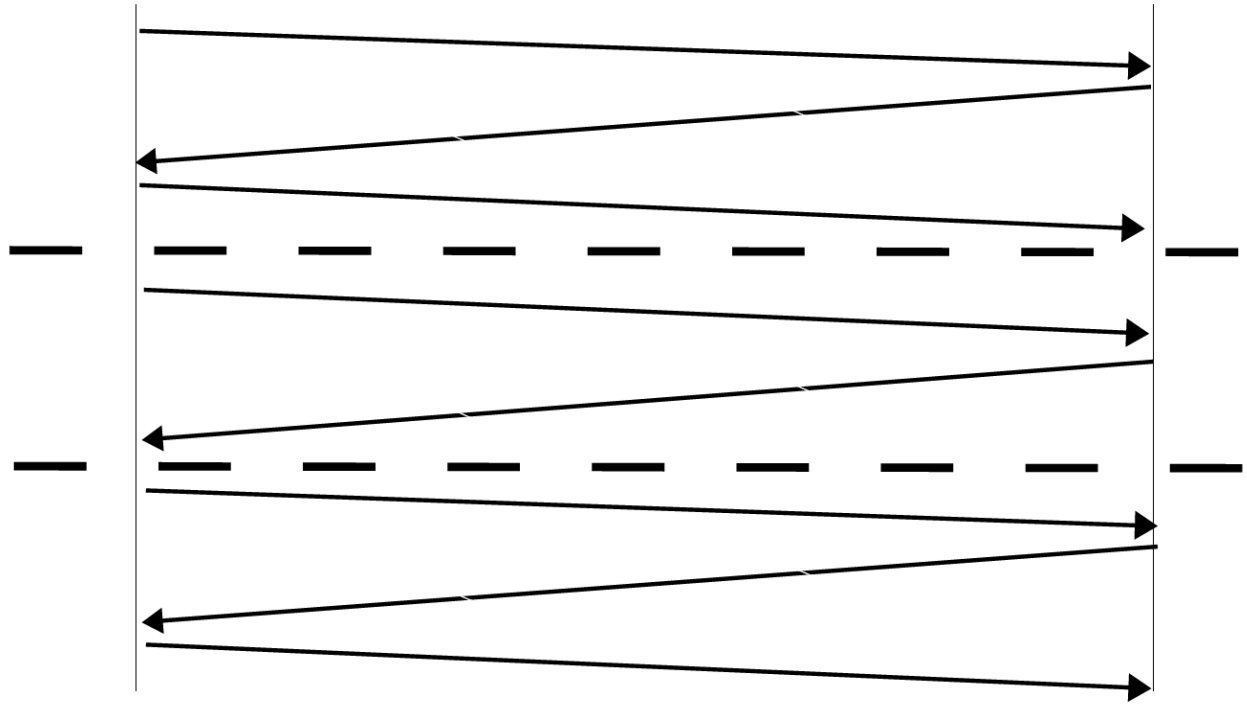
3. Utilisez **telnet** ou **netcat** comme client TCP. Remarque : **telnet** ou **netcat** sont des programmes capables d'ouvrir une socket TCP sur un port et de dialoguer en mode texte (ASCII).

   `telnet localhost 5000` ou `nc -vv localhost 5000`

4. Réalisez quelques échanges de textes entre le client et le serveur. Quittez une session telnet, faire `CTRL+]`, puis taper `quit`

5. Arrêtez la capture wireshark et analysez les trois étapes : ouverture de connexion, dialogue et fermeture de connexion (les flags associés et les numéros de séquence seq et ack).

Client Num port :	Serveur Num port :



Remarque : indiquer sur chaque flèche les flags (SYN, ACK, …) ainsi que que les numéros de séquence (seq et ack).

6. Exécutez un serveur TCP avec **netcat** sur un des ports compris entre 5000 et 5005 sur votre machine et détecter les ports ouverts acceptant des connexions TCP dans la plage 5000-5005, en utilisant l'outil **nmap** ou **netcat**.

   `nmap -sT -p 5000-5005 localhost` ou `netcat -vv -z localhost 5000-5005`

7. Faite de même qu'au point 6 mais dans la plage 7-13. Que constatez-vous ?

8. Exécutez simultanément sur votre VM deux programmes serveur sur le même port. Est-il possible d'exécuter deux serveurs TCP sur le même port ?

9. Essayez de lancer deux clients telnet ou netcat et faites quelques échanges avant de fermer vos deux connexions. Que se passe-t-il ?

1.2. Exercice 2 – UDP

10. Exécutez `netcat` en mode serveur UDP sur le port 5000.

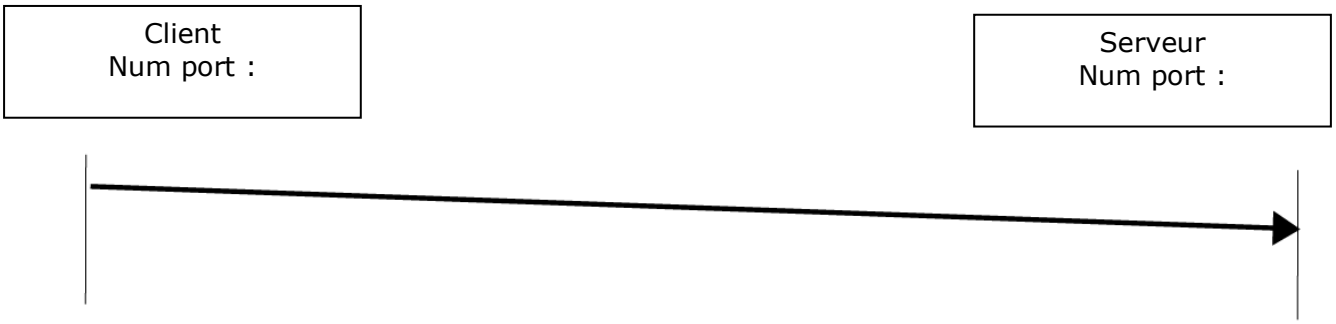
    `nc -l -p 5000 -u`

11. Réalisez une capture des échanges entre le serveur et le client et décodez le niveau UDP (utiliser l'annexe sur le protocole UDP).

12. Utilisez `netcat` comme client UDP.

    `nc -u 192.168.254.5 5000`

13. Arrêtez la capture wireshark et analysez.

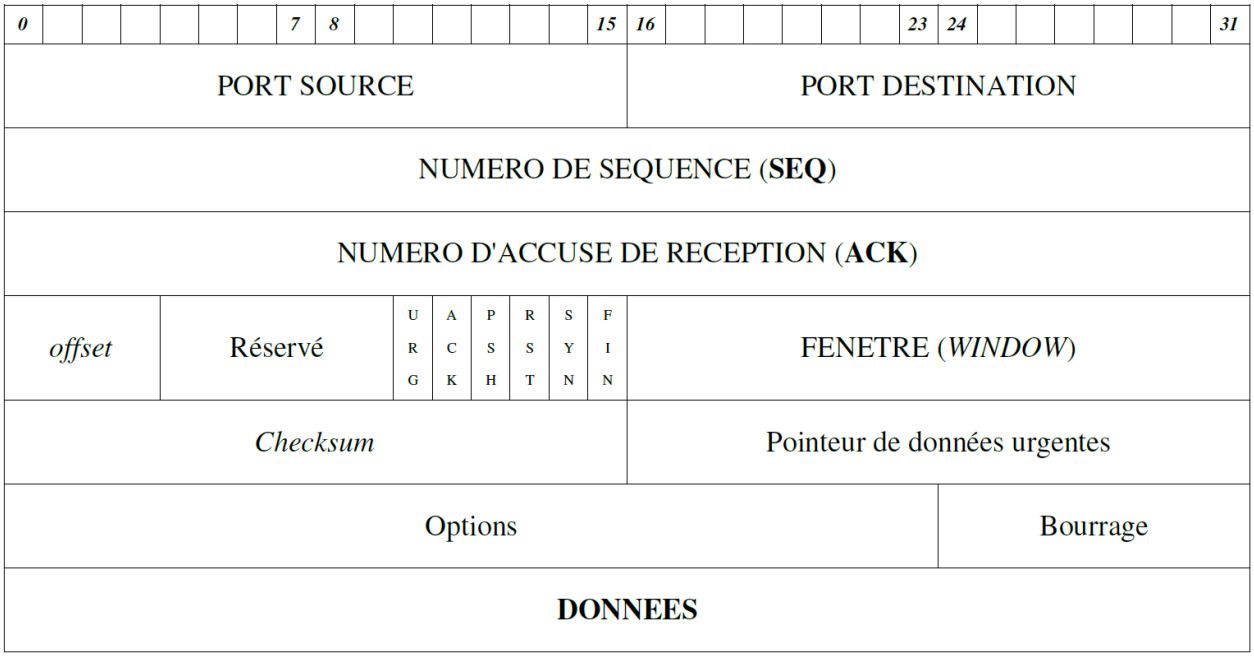


14. Est-il possible d'exécuter deux serveurs UDP sur le même port ? Testez.

15. Est-il possible d'exécuter un serveur TCP et un serveur UDP sur le même port ? Testez.

2 ANNEXES

2.1. Description d'un datagramme TCP



2.1.1 Description

Ports Source et Destination : correspond au port relatif à l'application (et à son protocole de couche application) en cours sur la machine source et destination, codés sur 16 bits (voir /etc/services)

SEQ (*SeQuence Number*) (32 bits) : compte les octets du flux de transmission de manière à identifier la position du premier octet de données d'un segment dans le flot des données initiales

ACK (*ACKnowledgment number*) (32 bits) : contient le n° de séquence du prochain octet attendu

Offset (ou longueur de l'entête) (4 bits) : le déplacement en mots de 32 bits (4 octets) du début des données de l'application.

Flags : **URG** (Urgent Pointer) : contient des données urgentes
ACK (Acknowledge Field) : acquittement
PSH (Push Flag) : passer immédiatement les données à la couche application
RST (Reset Flag) : forcer la clôture d'une connexion après une erreur irrécupérable.
SYN (Synchronize Flag) : synchroniser le démarrage d'une connexion
FIN : pour terminer une connexion

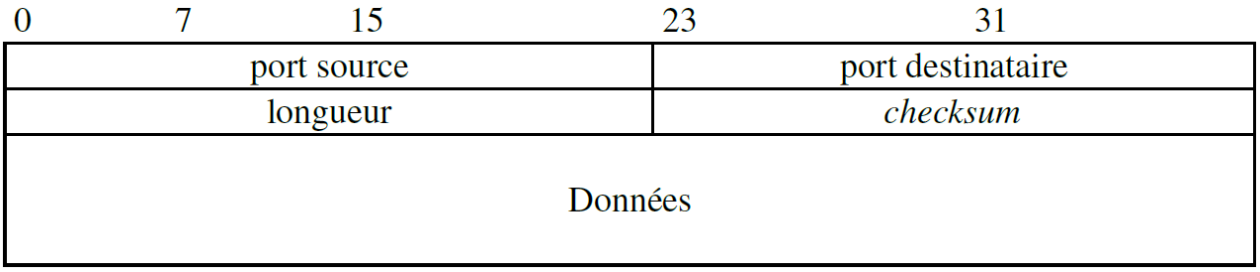
Window (Fenêtre) (16 bits) : annonce le nombre d'octet que le récepteur peut accepter (cf. contrôle de flux)

Checksum (16 bits) : contient une somme de contrôle de l'entête (cf. IP)

Urgent Pointer (Pointeur de données urgentes) (16 bits) : pointe à la fin d'un champ de données considéré comme urgent.

Options (longueur variable) : par exemple le MSS (Maximum Segment Size) désignant la taille maximum du segment à envoyer + ***Padding*** : habituellement rempli de 0 de manière à aligner le début des données sur un multiple de 32 bits.

2.2. Description d'un datagramme UDP



UDP permet à une application d'envoyer des messages à une autre application avec un minimum de fonctionnalités (pas de garantie d'arrivée, ni de contrôle de séquencement).

UDP n'accepte pas de datagramme de taille supérieure à 8KO.

UDP permet simplement d'utiliser les numéros de port.