**Protocoles TCP/IP - Corrigé des TP**

Analyse des protocoles TCP-IP

Vous allez pour cette partie utiliser l’analyseur de protocoes Wireshark : <http://www.wireshark.org/>. Son fonctionnement est décrit dans le TP du chapitre 2.

**1** Analyse de la commande ping.

* Quel protocole de niveau réseau est utilisé par la commande Ping ?
Le protocole ICMP qui permet de contrôler le fonctionnement d’IP.
* Effectuez un ping vers votre passerelle. Utilisez un filtre d’affichage permettant de visualiser uniquement les trames correspondant à une commande ping. Analysez les trames visualisées. Combien trouvez-vous de trames ? A quoi correspondent-elles. Quels sont les protocoles présents dans une trame ping ?
Pour le filtre, il suffit de taper icmp dans la fenetre « Filter »
Vous devez trouver en principe 8 trames ICMP : 4 demandes d’echo (*Echo Request*) et les 4 réponses correspondantes (*Echo Reply*). Le programme ping répète par défaut 4 fois le test de connectivité pour limiter les aléas.
Le message ICMP est encapsulé dans un paquet IP lui-même encapsulé dans une trame Ethernet ou WiFi.
* Quels sont les codes ICMP permettant de déterminer la nature du message ICMP (*Echo Request* ou *Echo Reply*).
Code 8 pour *Echo Request*, code 0 pour *Echo Reply*
* A quoi correspondent les données (« Data ») ICMP ?
Par défaut, le protocole ICMP envoie un bloc de données de 32 octets (généralement les lettres de l’alphabet) dans un message *Echo Request*. La réponse (*Echo Reply*) doit contenir le même bloc, ce qui confirme la connectivité IP entre les deux machines.

**2**Analyse de la commande Tracert

* Lancez une capture. Effectuez une commande **tracert** sur un serveur web de votre choix.  Quel protocole de niveau réseau est utilisé par la commande tracert ? Utilisez un filtre d’affichage permettant de visualiser uniquement les trames correspondant à une commande tracert.
La commande **tracert** utilise également le protocole ICMP qui permet de contrôler le fonctionnement d’IP.

* Observez les champs TTL des trames émises par votre machine et le champ Type des trames reçues. A partir de ces observations, expliquez le fonctionnement de la commande tracert.
Le programme **tracert** commence par envoyer un message *Echo Request*avec une valeur de TTL à 1. Le premier routeur rencontré décrémente le TTL qui passe à zéro. Le routeur supprime donc le paquet en revoyant un avertissement au PC source de type *Time-to-live Exceeded*. Le PC source peut donc ainsi connaitre le temps aller-retour jusqu’au premier routeur. Pour limiter les aléas, ce processus est répété 3 fois.
La recherche de route continue avec trois nouveaux messages ICMP *Echo Request*comportant une valeur de TTL à 2. Le deuxième routeur sera atteint et reverra à son tour un message *Time-to-live Exceeded.*La route etle temps aller-retour jusqu’au deuxième routeur seront connu.
Le processus se poursuit jusqu’à la machine destination, en incrémentant au fur et a mesure le TTL du paquet IP. La machine destination qui reconnaitra son adresse renverra finalement un message ICMP *Echo Reply.*

**3** Analyse du protocole IP

* Lancez une capture dans laquelle le protocole IP est présent (vous pouvez utiliser un filtre d’affichage pour isoler certaines trames).
* Quelles sont les couches OSI présentes dans les trames relevées ?
* Quelle est la version de IP ?
* Quelle est la taille l’en-tête IP ?
* Le paquet IP est –il fragmenté ? Justifiez votre réponse.
* Quelle est la valeur du champ TTL de l’en-tête IP ? Justifiez votre réponse.
* Quel est le protocole de niveau supérieur ? Par quelle valeur est-il identifié ?
* Pouvez-vous retrouver les valeurs hexadécimales des adresses IP source et destination ?
Dans l’exemple ci-dessous, la capture correspond à une application de relevé de message à l’aide du protocole POP3. On voit apparaitre les 4 couches correspondant aux 4 protocoles : Ethernet, IP, TCP et POP.
L’analyse du protocole IP est développée. La fenêtre du bas donne les valeurs en hexadécimal des 20 octets constituant l’en-tête IP (les 20 octets analysé au-dessus). Les adresses IP source et destination correspondent aux 8 derniers octets.

On peut vérifier que 212.227.15.140décimal = d4.e3.0f.8chexa et que 10.1.54.125décimal = 0A.01.36.7dhexa



**4** Analyse d’une connexion TCP.

* Lancez une capture. Connectez-vous à un site web quelconque, puis déconnectez-vous. Analysez la connexion TCP à l’aide de filtres d’affichage.
* Combien y-a-t-il de segments d’ouverture et de fermeture de la connexion ? Quelles sont les valeurs des « flags » pour ces segments ?
3 segments d’ouverture de la connexion :
Demande d’ouverture, SYN = 1 : ACK = 0
Acquittement d’ouverture : SYN = 1, ACK = 1
Confirmation d’ouverture : SYN = 0, ACK = 1

4 segments de fermeture de la connexion :
Demande de fermeture de la source : FIN = 1, ACK = 1
Acquittement de fermeture de la destination : FIN = 0, ACK = 1
Demande de fermeture de la destination : FIN = 1, ACK = 1
Acquittement de fermeture de la source : FIN = 0, ACK = 1
* Quels sont les numéros de port utilisés. Justifiez votre réponse.
Pour un protocole applicatif http, le port destination dans un segment destiné au serveur web doit être égal à 80. Le port source destiné à référencé la connexion sur le PC est supérieur à 1024.
* Vérifiez l’évolution des numéros de séquence et des numéros d’acquittements pour deux trames successives en réponse l’une à l’autre. Vous pouvez pour cela vous aider d’un schéma.
Dans l’exemple ci-dessous, le numéro d’acquittement renvoyé par la station permet d’acquitter les 1366 octets émis par le serveur à partir du numéro de séquence 1346159403 (1346159403+1366=1346160769).
Dans l’autre sens de transmission, le numéro d’acquittement envoyé par le serveur indique qu’il est prêt à recevoir les octets suivants à partir du numéro 2999255511, la station répond avec ce numéro de séquence.





* Wireshark possède un outil nommé, *Follow TCP stream* (menu Statistics/Flow Graph/TCP Flow). Vous pouvez ainsi visualiser l’enchainement des segments avec les du numéro de séquence et d’acquittement pour toute la session TCP.