EI-2I TP3

POLYTECH SORBONNE

1. PING et ICMP

Nous allons, lors de ce TP, étudier l'outil d'Internet **Ping** et **ICMP** à l'aide aux deux liens:
https://en.wikipedia.org/wiki/Ping_(networking_utility)
https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol

Nous allons nous intéresser à l'étude de traces (trames Ethernet) capturées pendant l'exécution de commandes ping. Les captures ont été effectuées au niveau de la machine locale. Les traces récupérées ont été enregistrées.

Installation de **WireShark** est nécessité sur votre machine pour réaliser ce TP. Un fichier pré-capturé vous est fourni sur Moodle.
Commencez donc par récupérer le fichier suivant et à le sauvegarder sur votre compte : **capture_ping.pcap**

La commande **ping** permet l'envoi de requêtes ICMP de type **ECHO**. Pour rappel, **ECHO** et **ECHO_REPLY** sont utilisés pour vérifier l'état d'activité d'une machine. Une machine source envoie un message **ECHO** à la machine destinataire dont elle veut vérifier l'activité. Celle-ci doit alors lui répondre par un message **ECHO_REPLY**. Pour plus d'informations sur cette commande et ses paramètres, consultez le manuel d'utilisation (commande **man ping** sur votre terminal).

1.1. Lancez l'application Wireshark et ouvrez le fichier de capture capture_ping.pcap. Appliquez le filtre suivant : ip.proto == 1. Que signifie ce filtre ?

Au début de l'expérimentation, nous avons exécuté la commande suivante :

```
ping -c 4 ufr-info-p6.jussieu.fr
```

Cette commande permet l'envoi de 4 requêtes ICMP de type **ECHO** à la machine distante de nom **ufr-info-p6.jussieu.fr.** Cette dernière répond à chacune des requêtes par un message ICMP de type **ECHO_REPLY**. Les trames relatives à ces requêtes/ réponses doivent être les 8 premières trames affichées dans votre fenêtre Wireshark (après application du filtre).

1.2. À l'aide du manuel d'utilisation de la commande ping, vérifiez la signification des paramètres présents dans la commande exécutée.

Dans Wireshark, sélectionnez **la première trame** et affichez les différents champs des entêtes Ethernet, IP et ICMP (fenêtre du milieu).

1.3. Quelle est l'adresse IP de la machine source (initiatrice des requêtes ECHO) ? Sachant que le masque de sous-réseau est le suivant : 255.255.255.0, quelle est l'adresse du sous-réseau hébergeant la machine source ?

1.4. Quelle est l'adresse IP de la machine distante (ufr-info-p6.jussieu.fr) ? Sachant que le masque de sous-réseau est le suivant : 255.255.255.0, quelle est l'adresse du sous-réseau hébergeant la machine distante ? Les deux machines se trouvent-elles dans le même sous-réseau ?

1.5. Quelle est l'adresse MAC source de la trame ? À quelle machine correspond cette adresse ?

1.6. Quelle est l'adresse MAC destination de la trame ? À quelle machine correspond cette adresse ?

1.7. Est-ce que ce datagramme IP a été fragmenté ? Expliquez.

1.8. Quelle est la longueur de l'entête IP ? Ce paquet contient-il des options ?

1.9. Quelle est la longueur du champ de données (champ data) du datagramme IP ? Comment avez-vous obtenu cette valeur ?

1.10. Quelle est la longueur de l'entête ICMP ? Identifiez et expliquez les différents champs de cet entête.

1.11. Quelle est la longueur des données ICMP véhiculées dans ce message ? À l'aide du manuel d'utilisation de la commande ping, vérifiez le nombre, par défaut, d'octets envoyés.

Dans Wireshark, sélectionnez **la deuxième trame** (correspondant à un ECHO_REPLY) et affichez les différents champs des entêtes IP et ICMP (fenêtre du milieu).

1.12. Quelle est la longueur de l'entête IP ?

1.13. Quelle est la longueur du champ de données (champ data) du paquet IP ?

1.14. Comparez les différents champs de l'entête ICMP avec ceux de la trame précédente. Expliquez les similitudes et les différences constatées.

1.15. Comparez les données véhiculées par ce message ICMP (code hexadécimal dans la fenêtre du bas) avec ceux de la trame précédente.

1.16. Comparez les différents champs des entêtes IP et ICMP des 4 trames de type ECHO (trames 9, 83, 206 et 344). Expliquez les différences constatées.

Nous avons continué l'expérimentation en exécutant la commande suivante :

```
ping -c 1 -s 2000 www.lip6.fr
```

Cette commande permet l'envoi d'une requête ICMP de type **ECHO** contenant 2000 octets de données à la machine distante de nom **www.lip6.fr.** La machine distante répond à cette requête par un message ICMP de type **ECHO_REPLY**. Dans votre fenêtre Wireshark, les trames relatives à cet échange doivent être positionnées juste après les 8 trames étudiées précédemment.
Pour info, une trame Ethernet ne peut contenir plus de 1500 octets de données. C'est donc la taille maximale d'un paquet IP dans un réseau Ethernet. De ce fait, un message **ECHO** contenant 2000 octets de données sera systématiquement fragmenté.

Dans Wireshark, sélectionnez **la première trame de ce nouvel échange** (la neuvième trame qui apparaît dans la fenêtre du haut quand le filtre **ip.proto == 1** est appliqué) et affichez les différents champs de l'entête IP (fenêtre du milieu). Cette trame correspond au premier fragment du message **ECHO**.

1.17. Dans l'entête IP, quelle information indique que ce datagramme a été fragmenté ?

1.18. Dans l'entête IP, quelle information indique que c'est le premier fragment ?

1.19. Quelle est la longueur totale de ce datagramme (premier fragment) ?

1.20. Quelle est l'adresse IP de la machine distante (www.lip6.fr) ?

Dans Wireshark, sélectionnez **la deuxième trame de cet échange** et affichez les différents champs de l'entête IP (fenêtre du milieu). Cette trame correspond au deuxième fragment du message **ECHO**.

1.21. Dans l'entête IP, quelle information indique que ce n'est pas le premier fragment ?

1.22. Dans l'entête IP, quelle information indique que c'est le dernier fragment ?

1.23. Quelle est la longueur totale de ce datagramme (deuxième fragment) ?

1.24. Calculez la somme des longueurs de ces deux fragments. Expliquez la valeur obtenue.

1.25. En combien de fragments a été divisé le message ECHO_REPLY ? Quelle est la longueur totale des données véhiculées par ce message ?

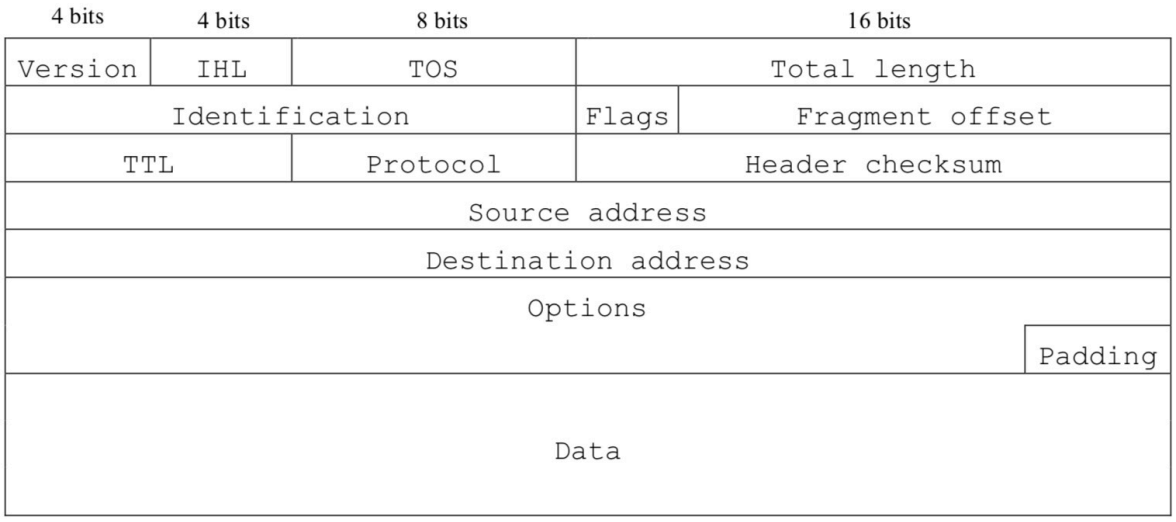
Finalement, nous avons terminé l'expérimentation en exécutant la commande suivante :

```
ping -c 1 -s ? rp.lip6.fr
```

Cette commande permet l'envoi d'une requête ICMP de type **ECHO** contenant un certain nombre d'octets de données à la machine distante de nom **rp.lip6.fr**. La machine distante répond à cette requête par un message ICMP de type **ECHO_REPLY**. La commande précédente est incomplète : la valeur du paramètre –s n'a pas été donnée. Dans votre fenêtre Wireshark, les trames relatives à cet échange doivent être positionnées juste après les 12 trames étudiées précédemment.

1.26. En combien de fragments a été divisé le message ECHO REQUEST ? Complétez la commande précédente en identifiant la longueur totale des données ICMP véhiculées par ce message.

DATAGRAMME IP



! `Version` indique le format de l'en-tête. Ce champ sert à l'identification de la version courante du protocole. La version décrite ici (et aujourd'hui utilisée) porte le n°4 ;

! `IHL` (*IP Header Length*) est la longueur de l'en-tête IP exprimée en mots de 32 bits ;

! `TOS` (*Type Of Service*) définit le type de service à appliquer au paquet en fonction de certains paramètres comme le délai de transit, la sécurité. Il est peu utilisé et sa valeur est généralement égale à 0 ;

! `Total Length` est la longueur totale du datagramme, exprimée en octets. En pratique, il est rare qu'un datagramme IP fasse plus de 1500 octets ;

! `Identification` sert en cas de fragmentation/réassemblage du datagramme. Ce champ permet alors à l'entité réceptrice de reconnaître les fragments issus d'un même datagramme initial et qui doivent donc faire l'objet d'un réassemblage ;

! `Flags` (3 bits) est utilisé par la fragmentation. Il est composé (de gauche à droite) :

– d'un bit réservé : mis à 0 ;

– de l'indicateur DF (*Don't Fragment*): mis à 1 par l'émetteur pour interdire la fragmentation ;

– de l'indicateur MF (*More Fragment*) : mis à 1 pour signifier que le fragment courant est suivi d'un autre fragment ;

! `Fragment offset` (13 bits) donne la position relative du fragment dans le datagramme initial, le déplacement étant donné en unités de 64 bits ;

! `TTL` (*Time To Live*) donne une indication de la limite supérieure du temps de vie d'un datagramme ;

! `Protocol` indique le protocole (de niveau supérieur) utilisé pour le champ de données du datagramme. Quelques exemples :

Code (déc)	Abréviation	Nom du protocole	Référence
1	ICMP	Internet Control Message Protocol	[RFC792]
2	IGMP	Internet Group Management Protocol	[RFC1112]
6	TCP	Transmission Control Protocol	[RFC793]
8	EGP	Exterior Gateway Protocol	[RFC888]
9	IGP	any private Interior Gateway Protocol	
17	UDP	User Datagram Protocol	[RFC768]
36	XTP	XTP	
46	RSVP	Reservation Protocol	
…	…	…	

! `Header Checksum` est une zone de contrôle d'erreur portant uniquement sur l'en-tête du datagramme ;

! `Source Address` est l'adresse IP de la source du datagramme ;

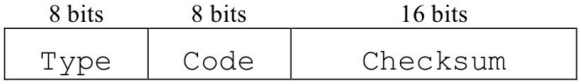
! `Destination Address` est l'adresse IP de destination du datagramme ;

! `Options` est de longueur variable. Il sert à des fonctions de contrôle utiles dans certaines situations. Il est constitué d'une succession d'options élémentaires, également de longueurs variables. Les options sont codées sur le principe TLV (Type, Longueur, Valeur). La longueur indique la taille complète de l'option en octets.

! `Padding` est de longueur variable : il permet d'aligner l'en-tête sur 32 bits.

MESSAGE ICMP

Les messages ICMP sont encapsulés dans des datagrammes IP. Ils ont tous en commun le même format pour le premier mot de 32 bits :



Type	Message	Objet
0	Echo Reply	Réponse en écho
3	Destination Unreachable	Destination inaccessible
4	Source Quench	Interruption de la source
5	Redirect	Redirection, changement de route
8	Echo	Demande d'écho
11	Time Exceeded	Temps de vie d'un datagramme dépassé
12	Parameter Problem	Datagramme mal formé
13	Timestamp	Demande de date d'estampillage
14	Timestamp Reply	Réponse à une demande d'estampillage
15	Information Request	Demande d'information
16	Information Reply	Réponse à une demande d'information
17	Address Mask Request	Demande de masque d'adresse
18	Address Mask Reply	Réponse à une demande de masque d'adresse

A titre d'exemple, Echo et Echo Reply sont utilisés pour vérifier l'état d'activité d'une machine. Une machine source envoie alors un message Echo à la machine destinataire dont elle veut vérifier l'activité. Celle-ci doit alors lui répondre par un message Echo Reply. L'adresse source dans un Echo (Type = 8) sera l'adresse destinataire du Echo Reply (Type = 0). Pour constituer un Echo Reply, les adresses source et destination sont donc simplement inversées. Les données reçues dans un Echo doivent être retournées dans le Echo Reply. Deux champs du message, Identifier et Sequence Number, sont utilisés par l'émetteur de l'Echo pour mettre en correspondance les réponses avec les requêtes. Par exemple, l'identificateur peut correspondre à un port TCP ou UDP pour identifier une session et le numéro de séquence être incrémenté pour chaque requête d'écho émise. Le répondeur retourne les mêmes valeurs dans sa réponse.