

Ensimag 1^{ère} année

TP n°1 — Observation d'Internet et anatomie d'une application Web

Il est recommandé de prendre des notes.

Une question sur l'effet ou l'utilisation d'une commande?

man nom_de_la_commande!

1 Environnement

Les séances de TP ont lieu dans les salles informatique de l'établissement. Ces salles sont équipées de machines nommées `ensipcXYZ` (où XYZ désigne un numéro de machine). Ces machines possèdent, entre autre, une interface réseau qui leur permet de communiquer entre elles, avec les serveurs de l'école comme `pcserveur.ensimag.fr`, et aussi avec Internet.

Chaque étudiant a à sa disposition une machine, sur laquelle il peut observer précisément le trafic réseau émis et reçu par l'interface réseau de la machine. Notez que d'autres utilisateurs peuvent être connectés (à distance) à la machine que vous utilisez. Ceci peut être problématique pour deux raisons : (1) les trafics engendrés par chacun des utilisateurs d'une station partagée sont mélangés, ce qui perturbe les possibles observations, et (2) problèmes de confidentialité pour certains protocoles n'utilisant pas de chiffrement. Dans la mesure du possible, choisissez donc une machine où personne n'est connecté.

Pour les étudiants devant réaliser le TP à distance pour une raison valable, il est fortement conseillé de se connecter à distance à une des machines de l'école pour avoir le bon environnement réseau et logiciel. Pour cela (au moins) deux solutions possibles :

- VPN + RDP (par exemple avec Microsoft Remote Access, ou Remmina qui gère mieux les caractères spéciaux dans les mots de passe) ou
- VPN + SSH + `vnc_web`

Pour démarrer :

- Démarrez les machines (sous « Linux Ubuntu ») si ce n'est pas déjà fait.
- Identifiez-vous à l'aide de vos identifiants Ensimag.

2 Réseau Internet

Le réseau Internet est composé de nombreuses stations, appelées hôtes, interconnectées par des liens de transmission et des équipements intermédiaires appelés routeurs. L'échange d'informations entre la source et la destination s'effectue à l'aide du protocole IP (Internet Protocol). Un paquet IP spécifie l'adresse source et destination, et les routeurs intermédiaires se chargent d'acheminer le paquet à destination. Pour pouvoir manipuler facilement des adresses, le réseau maintient la correspondance entre une adresse IP (par exemple 129.88.47.4) et un nom symbolique (par exemple `delos.imag.fr`). Cette fonction est assurée par le système de nommage DNS (Domain Name System) qui sera étudié plus en détail lors d'un prochain TP.

2.1 Prise en main de l'outil Wireshark

Wireshark est un puissant analyseur de trafic réseau. Il observe tous les messages qui sont émis et reçus sur une interface réseau de la machine. Cette partie du TP vous permettra de prendre en main cet outil pour en maîtriser les fonctionnalités de base qui seront utilisées dans ce TP et les suivants.

2.1.1 Le point sur les interfaces

Une *interface réseau* est une entité permettant à une machine de communiquer avec d'autres. Le plus souvent, elle prend la forme d'un périphérique matériel capable d'envoyer et de recevoir des messages, comme les *cartes Ethernet* communément appelées *cartes réseaux* par abus de langage. Elle peuvent aussi être *logicielles* ou *virtuelles* : dans ce cas, elle ne correspondent pas à un périphérique matériel installé sur la machine, mais à un élément logiciel du système d'exploitation. C'est le cas par exemple de la *boucle locale*, souvent nommée `lo0` ou `lo`, une interface logicielle activée par le système pour capturer le trafic interne à une machine. Par exemple, la boucle locale peut être utile pour tester facilement une application censée envoyer et recevoir des messages sur un réseau, lors de sa mise au point.

Q 1 — *Il existe, sur les machines des salles de l'Ensimag, plusieurs interfaces réseaux. Listez-les en exécutant la commande `ifconfig` dans un terminal.*

Vous devriez voir apparaître plusieurs interfaces. Certaines comme `en0<numbers>` ou `enp<numbers>s<numbers>` sont des interfaces physiques (carte réseau), `lo` est une interface virtuelle correspondant à la boucle locale, `virbr0` est également une interface virtuelle. Dans certaines salles les machines sont équipées d'une autre interface physique qui n'est pas configurée : vous remarquerez par ailleurs qu'elle ne dispose pas d'**adresse IP**.

Notez que sur d'autres machines, l'interface `enp<numbers>s<numbers>` peut avoir un autre nom, comme par exemple `em0` ou `eth0`.

2.1.2 Première analyse de trafic

Certaines fonctionnalités de Wireshark nécessitent des privilèges particuliers pour pouvoir fonctionner. Vos machines ont été configurées avec les privilèges nécessaires pour autoriser la capture ;

mais sur votre machine personnelle, vous aurez sûrement besoin des droits de root.

L'outil **Wireshark** se lance donc depuis le terminal en tapant la commande :


```
wireshark &
```


ou, si un message d'erreur vous informe que vous avez besoin des droits root :

```
sudo wireshark &
```

Notez que le caractère & ne fait pas partie du nom de la commande, il sert simplement à lancer l'outil en arrière-plan (cf. *poly UNIX*).

Nous allons dans un premier temps utiliser Wireshark pour analyser le trafic réseau (les messages échangés et leur contenu) généré par le chargement d'une page web hébergée sur Internet.

Q 2 — Lancez Wireshark dans un terminal, et démarrez une capture sur l'interface réseau *Ethernet eno1* [. . .]. Pour ce faire, cliquez sur l'icône  (show the capture options). Choisissez l'interface à capturer et cliquez sur le bouton start. Utilisez ensuite le navigateur web de votre choix pour charger une page de votre choix puis arrêtez la capture.

Vous pouvez aussi arrêter, lancer et relancer une capture à partir des icônes  dans la barre des menus. La fenêtre principale de Wireshark doit maintenant contenir un grand nombre d'entrées (ne prenez pas peur! Nous les filtrerons pour y voir plus clair).

Chaque entrée correspond à un échange enregistré par la capture sur l'interface choisie. Vous pouvez explorer le contenu de chaque échange en cliquant sur l'entrée correspondante. Vous verrez apparaître le contenu des messages échangés. Ce contenu s'affiche en ligne, et à chaque ligne correspond une couche du modèle OSI. Wireshark représente ainsi l'encapsulation du message à travers les différentes couches. Notez que l'ordre d'affichage est inversé par rapport aux représentations habituelles du modèle OSI : chaque couche listée encapsule les couches situées en dessous.

2.1.3 Petite désencapsulation à la main

Wireshark enregistre l'ensemble des informations transitant par une interface donnée, ce qui, comme vous pouvez le constater, rend la visualisation de ces informations fastidieuses. C'est pourquoi Wireshark est muni d'un mécanisme de filtrage permettant d'afficher uniquement les informations souhaitées.

Par exemple, lorsque vous utilisez un poste de travail connecté au réseau de l'Ensimag, les protocoles LDAP et NFS (couche applicative du modèle OSI) sont utilisés pour vous authentifier sur le réseau et synchroniser vos fichiers entre votre poste et le serveur de fichiers, pour que vous puissiez les retrouver à votre prochaine connexion sur un autre poste.

Voici en exemple deux filtres permettant d'enlever les trames liés à ces protocoles que nous ne regarderons pas dans ce TP, pour faciliter vos expérimentations :

`!ldap && !nfs`

L'opérateur `&&` correspondant au « et » logique, et `!` correspondant à la négation logique.

Les filtres sont à entrer dans la barre de texte `Apply a display filter` au dessus de la liste des trames. **N'oubliez pas d'appuyer sur "Entrée" pour valider les filtres.**

Q 3 — *Utilisez, ou rajoutez au précédent filtre, un filtre « `tcp` » pour voir les informations correspondant à votre précédente connexion au web.*

Remarquez que vous filtrez ici l'affichage des entrées, en n'affichant que celles dont un des champs contient « `tcp` ».

Q 4 — *Comparez le nombre de trames (`Packets` pour Wireshark) capturées au nombre de trames affichés (`Displayed`) suite au filtrage.*

Vous avez accès à la liste des filtres existants classés par protocoles, en faisant un clic-droit sur le champ de saisie des filtres puis `Display Filter Expression`. Cette fenêtre contient une barre de recherche. La liste au dessus est triée par protocole.

Q 5 — *Construisez un filtre qui permet d'afficher uniquement les requêtes du protocole TCP contenant le flag « `syn` » et vérifier qu'il fonctionne.*

Attention cependant, une fois activé, le filtre s'appliquera à toutes les captures suivantes. Pour rétablir l'affichage de toutes les entrées (sans filtre), utilisez le bouton `[X] Clear display filter` à droite du champ de texte.

Lors du chargement d'une page web, le processus d'encapsulation de chaque couche ajoute son propre en-tête aux données de votre page. En cliquant sur une ligne de Wireshark, vous allez pouvoir sélectionner un paquet. Sélectionnez un paquet TCP SYN en utilisant le filtre mis en place précédemment.

Q 6 — *Dépliez la couche Transmission Control Protocol (TCP) et repérez le flag SYN.*

Q 7 — *Quelles sont les différentes couches utilisées pour envoyer un paquet TCP de type SYN? Repérez les différentes caractéristiques des couches vues en cours (champs des en-têtes IP/TCP).*

Q 8 — *Trouvez la longueur totale en octets de la trame et des paquets (PDU) de chaque couche, en détaillant pour chaque paquet : taille de l'en-tête & taille de la charge utile (données). Wireshark affiche certaines de ces données dans le détail des couches : cherchez les champs `Total Length`, `Header Length`. Vous pourrez en déduire les données manquantes en vous rappelant que chaque paquet contient dans sa charge utile le(s) paquet(s) qu'il encapsule.*

Wireshark ne prend pas en compte le préambule et le FCS (Frame Check Sequence) du paquet Ethernet, qu'il faudra donc rajouter.

Q 9 — *Que constatez-vous concernant la taille de la charge utile d'un paquet TCP de type SYN? Pourquoi?*

2.2 Test de la connectivité

Considérons le site WWW de l'Université de Californie à Berkeley ayant pour nom DNS `www.berkeley.edu`. Lancez une capture Wireshark et testez la connectivité entre votre station et ce site grâce à l'utilitaire `ping` afin de répondre aux questions qui suivent :

Q 10 — *Quel est le protocole utilisé par `ping` ? (cf. cours) Filtrez la capture Wireshark sur le protocole en question et expliquez sommairement le fonctionnement de `ping`.*

Q 11 — *Quelle est l'adresse IP du site web de Berkeley qu'utilise `ping` ? Filtrez la capture Wireshark par l'adresse trouvée.*

Q 12 — *Quel est le temps aller-retour entre votre station et le site de l'Université de Berkeley ?*

Q 13 — *Quel est le temps aller-retour entre votre station (`ensipcXXX`) et celle de l'un de vos voisins de TP ?*

2.3 Analyse des itinéraires

Les paquets de test envoyés par `ping` sont acheminés par un ensemble de routeurs à travers le réseau. On peut mesurer le temps aller-retour d'un paquet vers les routeurs intermédiaires à l'aide de l'utilitaire `tracert`.

Q 14 — *Utilisez `tracert` pour déterminer le nombre de routeurs traversés lors d'une communication entre votre station et le serveur `pcserveur.ensimag.fr`. Que peut-on en conclure sur la connexion entre vos machines et ce serveur ?*

Q 15 — *Lancez une capture Wireshark, puis lancez `tracert` et déterminez le nombre de routeurs entre votre station et `intranet.u-ga.fr`.*

Nous allons maintenant nous servir de Wireshark pour comprendre le fonctionnement de l'outil `tracert`. L'objectif va être ici de mettre en évidence les échanges de messages entre votre machine et les routeurs traversés. A partir de la dernière capture Wireshark réalisée, filtrez l'affichage de manière à n'afficher que les échanges qui concernent `intranet.u-ga.fr`. Une façon simple d'y parvenir est de mettre en place un filtre d'affichage sur l'adresse IP utilisée par `tracert` pour contacter `intranet.u-ga.fr`, mise en évidence sur la première ligne du résultat de `tracert`. Entrez par exemple la chaîne de caractères : `(ip.addr == 195.83.24.194)` dans le champ de texte réservé aux filtres d'affichage pour n'afficher que les échanges où l'adresse IP 195.83.24.194 intervient.

Q 16 — *Quels sont les protocoles impliqués dans le fonctionnement de `tracert` ?*

Q 17 — *À quelle machine `tracert` envoie-t-il des messages ?*

Q 18 — *Qui répond et pourquoi ? Pour répondre à cette question, explorez le contenu des messages envoyés avec Wireshark, en remarquant en particulier l'évolution du champ TTL des paquets IP.*

Rappel : le *Time-To-Live* ou TTL est un champ de l'en-tête IP qui représente la durée de vie d'un paquet IP, sous la forme d'un nombre de routeurs pouvant être traversés avant la destruction du paquet (nombre de sauts). Ainsi, à chaque traversée d'un routeur, le champ TTL est décrémenté de 1. Le routeur qui décrément le TTL à 0 jette le paquet, et en informe la source du paquet.

Q 19 — *Faites un traceroute vers le site web `www. slac. stanford. edu` au États-Unis. `traceroute` parvient-il à révéler l'identité de tous les routeurs ?*

Q 20 — *Entre quels routeurs traverse-t-on l'Atlantique (regardez les temps intermédiaires) ?*

Q 21 — *`mtr` est un outil de traceroute plus moderne que le vénérable `traceroute`. Utilisez `mtr` pour tracer la route vers `www. slac. stanford. edu`. (appuyez ensuite sur la touche `q` pour quitter). Quelle différence avec `traceroute` pouvez-vous observer dans Wireshark ?*

Un certain nombre de serveurs, répartis dans le monde, proposent de tracer la route entre eux et une destination quelconque, et présentent le résultat à l'utilisateur (`traceroute` inversé). Vous pouvez accéder à la liste sur le site web `www. traceroute. org`.

Q 22 — *Utilisez le site hébergé par Stanford aux États-Unis pour comparez le chemin aller et le chemin retour entre votre ordinateur et ce serveur (`www. slac. stanford. edu`).*

3 Anatomie d'une Application Web (WWW)

Nous analysons maintenant la façon dont des applications utilisent le réseau, en prenant comme exemple la navigation sur le World Wide Web (lisez la page wikipedia si la définition de *World Wide Web* – et par exemple la différence avec *internet* – n'est pas claire pour vous).

Quand un utilisateur clique sur un lien dans un document présenté par un navigateur web (par exemple Firefox) :

- le navigateur fait appel au protocole HTTP pour charger le document correspondant au lien ;
- HTTP ouvre une connexion TCP au niveau transport ;
- le protocole TCP utilise l'interconnexion au niveau IP pour échanger des segments de données avec le site Web distant ;
- enfin, IP utilise une connexion Ethernet sur le câble local pour dialoguer avec le routeur de raccordement de l'Ensimag au réseau extérieur.

Nous observerons les échanges de données à différents niveaux de protocoles au cours de l'accès à une page Web.

3.1 Analyse réseau

Après avoir lancé une capture Wireshark sur l'interface réseau Ethernet (vous pouvez filtrer pour observer uniquement les échanges utilisant le protocole HTTP), utilisez Firefox pour accéder à `http://moais.imag.fr/members.html`

Dans les questions suivantes, reliez les observations visuelles sur la page, et les échanges dans Wireshark au niveau du protocole HTTP. Utilisez le filtre « `http` ».

Q 23 — *Quels sont les échanges au niveau du protocole HTTP ?*

Q 24 — *Cherchez le paquet HTTP renvoyé par le serveur qui contient le code HTML de la page web dans sa charge utile (paquet dont le champ `Content-Type` de l'en-tête HTTP contient `text/html`). Trouver la manipulation dans Wireshark pour créer rapidement un filtre sur tous les paquets qui contiennent un contenu HTML.*

Q 25 — *Identifiez les balises HTML contenant les liens et les images inclus dans cette page.*

3.2 Exécution du protocole HTTP « à la main »

Maintenant, essayez d'appeler directement le protocole HTTP sans passer par le navigateur. Pour ce faire, récupérez la page `http://mois.imag.fr/members.html` en utilisant l'outil `telnet` avec le port 80 :

```
telnet mois.imag.fr 80
```

Telnet vous ouvre un dialogue direct avec le serveur HTTP de la machine spécifiée (le port 80 indique à la machine distante qu'elle doit vous mettre en relation avec le serveur qui comprend le protocole HTTP). Vous pouvez alors utiliser des commandes (plus précisément des PDU) du protocole HTTP, et vous commencerez par la commande `GET` pour demander une ressource au serveur. La commande prend deux arguments : le chemin relatif vers la ressource, et la version du protocole à utiliser.

Ici nous utiliserons `HTTP/1.1`, il faut donc en plus confirmer le nom de l'hôte à atteindre, sur une ligne qui suit la ligne du `GET` :

```
GET /members.html HTTP/1.1
Host: mois.imag.fr
```

(terminé par 2 retours à la ligne, donc avec une ligne vide). Observez l'en-tête et le corps de la réponse.

Q 26 — *Vérifiez que vous obtenez bien le code source de la page `http://mois.imag.fr/members.html`*

Q 27 — *Quelle est la réponse du serveur si on utilise la méthode `HEAD` au lieu de `GET`?*

Q 28 — *Quelle est la version du protocole utilisée par le serveur pour répondre?*

Q 29 — *Comparez maintenant les messages de connexion affichés sur le terminal lorsqu'on utilise `telnet` pour se connecter sur le port 80 de `mois.imag.fr` et de `lig-karok.imag.fr`. Que peut-on dire de ces deux noms d'hôtes?*

Pour confirmer ce que vous venez de conjecturer, on se propose de faire un `telnet` sur l'adresse IP de la question précédente, en récupérant la page d'accueil (`/`) avec la commande `GET` :

```
telnet <adresse ip> 80
...
GET / HTTP/1.1
Host: ...
```

avec la première fois l'hôte `lig-karok.imag.fr` puis renouvelez la commande avec l'hôte `mois.imag.fr`

Q 30 — *Obtenez-vous la même page? Pourquoi?*

4 Première observation des couches Réseau

Dans cette section, on utilise Wireshark pour observer les échanges à différents niveaux (Ethernet, IP, TCP) pendant le chargement d'une page web. Pour cela, utilisez une capture Wireshark du chargement dans Firefox de la page utilisée pour les questions précédentes (<http://moais.imag.fr/members.html>), en vous assurant de vider de cache si vous refaites une nouvelle capture (avec Firefox, Ctrl+Shift+R pour recharger une page en vidant le cache).

Filtrez la capture sur l'adresse IP du serveur trouvée précédemment. Identifiez le premier segment TCP qui initie la connexion (SYN) entre votre machine et le serveur HTTP.

4.1 Couche Ethernet

Q 31 — *Relevez les adresses MAC source et destination du paquet Ethernet correspondante. À quels hôtes correspondent-elles?*

4.2 Couche IP

Q 32 — *Quelles sont les adresses source et destination du paquet IP? À quels hôtes correspondent-elles?*

Q 33 — *Repérez le champ Protocole dans l'en-tête. À quel type de Protocole correspond-il?*

Q 34 — *Donnez les valeurs de ce champ Protocole pour un paquet ICMP et UDP (n'hésitez pas à regarder du côté des filtres Wireshark pour trouver ces types de paquets). Rappel : pour générer du trafic UDP et ICMP vous pouvez effectuer un traceroute vers un hôte.*

4.3 Couche TCP

Q 35 — *Quels sont les types d'échanges au niveau du protocole TCP (ceux qui contiennent du HTTP, ceux qui n'en contiennent pas...)?*

Q 36 — *Quels échanges TCP sont impliqués dans le chargement de votre page Web?*

Q 37 — *Identifiez les numéros de ports utilisés par le client pour ces requêtes. Pourquoi ces valeurs sont utilisées?*