

Ensimag 1^{ère} année

TP n°1 – Observation d’Internet et anatomie d’une application Web

Il est recommandé de prendre des notes. Se référer à la version numérique de ce document pour les liens (sur Chamilo).

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 d’un peu plus de 80 machines nommées **ensipcXYZ** (ou 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. Il vous est interdit de (vous n’avez pas les droits pour) réaliser de telles observations sur des machines sur lesquelles vous n’êtes potentiellement pas seuls. En effet, les trafics engendrés par chacun des utilisateurs d’une station partagée sont mélangés, ce qui perturbe les possibles observations. De plus, la confidentialité et la sécurité de vos activités ne seraient pas assurées dans le cas où n’importe quel utilisateur pouvait observer ce que fait l’autre.

Pour démarrer :

- Démarrez les machines en “**Linux (CentOS)**”, en validant ce choix dans le menu d’amorçage.
- Identifiez-vous à l’aide de vos identifiants Ensimag.

Attention : Si vous ne validez pas votre choix dans le délai imparti, la machine s’éteindra automatiquement (cela évite que des machines non-utilisées restent alimentées pour rien). Soyez donc prêts lorsque vous démarrez la machine, pour choisir Linux (CentOS) lorsque le choix s’affiche sur l’écran.

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 129.88.48.4) 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.

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. La première, `enp0s<numbers>`, est une interface physique (carte réseau), alors que la deuxième, `lo`, est une interface virtuelle correspondant à la boucle locale. Dans certaines salles les machines sont équipées d'une troisième 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 `enp0s<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 &
```

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.

1. Lancez Wireshark dans un terminal, et démarrez une capture sur l'interface réseau *Ethernet em1*[...]. 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 !). 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.

2.1.3 Petite désencapsulation à la main

Vous avez dû voir en cours que le modèle OSI fonctionne avec les données un peu comme une lettre que vous enverriez à un ami : vous allez l'écrire, puis vous la mettez dans une enveloppe et la timbrerez. La poste la mettra dans un sac de courrier, puis dans un camion. La poste destinataire la sortira du camion, puis du sac de courrier, et la déposera dans la boîte aux lettres de votre destinataire, avant que ce dernier ne l'ouvre et puisse lire votre courrier.

2. Trouvez depuis le cours des analogies entre les différentes couches protocolaires des réseaux et le service courrier.

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.

3. Utilisez 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". Vous avez accès à plusieurs filtres d'affichage prédéfinis en cliquant sur le bouton **Filter** : à gauche du champ de texte. Vous pouvez même créer de nouveaux filtres, basé sur des recherches en cliquant sur "Expressions" à droite du champ de recherche.

4. Construisez un filtre qui permet d'afficher uniquement les requêtes 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 **Clear** 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 packet TCP SYN en utilisant le filtre mis en place précédemment.

5. Quelles sont les différentes couches utilisées pour envoyer un packet TCP de type SYN ? Repérez les différentes caractéristiques des couches vues en cours (champs des en-têtes IP/TCP).
6. Notez la longueur totale du packet pour chaque couche et calculez la taille de l'en-tête. N'oubliez pas de prendre en compte que Wireshark ne prend pas en compte le préambule et le FCS (Frame Check Sequence).

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 :

7. Quel est le protocole utilisé par `ping` ? Filtrez la capture Wireshark sur le protocole en question et expliquez sommairement le fonctionnement de `ping`.
8. Quelle est l'adresse IP du site web de Berkeley qu'utilise `ping` ?
9. Quel est le temps aller-retour entre votre station et le site de l'Université de Berkeley ?
10. Quel est le temps aller-retour entre votre station (`ensipcXXX`) et celle de l'un de vos voisins de TP ?
11. Mesurez les temps minimaux, moyens, et maximaux pris pour transmettre des paquets contenant 500, 1000, 5000, 10000, et 25000 octets de données entre votre station et celle de votre voisin.
12. Ce temps dépend-il de la taille des paquets ? Justifiez votre réponse.
13. Trouvez une manière (il y en a plusieurs) de spécifier le nombre d'envois effectués par `ping`. Utilisez-la pour faire en sorte que `ping` s'arrête automatiquement après l'envoi et la réception d'une seule requête ?

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.

Par défaut, l'utilitaire `ping` envoie des paquets dont le champ TTL est initialisé à 64. On peut spécifier une valeur initiale de TTL avec l'option `-t` de `ping`.

14. On veut connaître le nombre de routeurs entre votre machine et le site web de IIJ, un fournisseur d'accès à Internet japonais : `www.iiij.ad.jp`. Pour cela, trouvez la plus petite valeur de `N` à partir de laquelle la commande `ping -t N www.iiij.ad.jp` s'exécute correctement. Déduisez-en le nombre de routeurs traversés entre votre machine et `www.iiij.ad.jp`.
15. A partir de la capture Wireshark d'une commande `ping` vers `www.iiij.ad.jp`, retrouvez la valeur du champ TTL d'un paquet IP contenant une réponse ICMP (echo reply) reçue par votre machine. Déduisez-en le nombre de routeurs traversés par ce paquet, et comparez le résultat obtenu à celui de la question précédente.

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`.

16. Utilisez **traceroute** 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 ?
17. Lancez une capture Wireshark, puis lancez **traceroute** pour déterminer le nombre de routeurs entre votre station et **intranet.u-ga.fr**. Déduisez de votre capture les protocoles impliqués dans le fonctionnement de **traceroute**.

La dernière commande exécutée devrait afficher une sortie ressemblant à ça :

```
traceroute to intranet.u-ga.fr (195.83.24.194), 30 hops max, 60 byte packets
 1 gateway (129.88.247.254)  1.074 ms  1.395 ms  1.520 ms
 2 simsu-bb-to-r-ensimag.ujf-grenoble.fr (152.77.32.213)  20.782 ms  20.765 ms  20.738 ms
 3 bio-bb-to-simsu-bb.ujf-grenoble.fr (152.77.39.30)  14.833 ms  14.832 ms  14.805 ms
 4 spring-tn-winter-leaf-111-to-bio-bb.u-ga.fr (152.77.0.206)  0.453 ms  0.686 ms  0.871 ms
 5 aci-bb-to-ksup.u-ga.fr (195.83.24.193)  0.640 ms  0.881 ms  1.101 ms
 6 aci-bb-to-ksup.u-ga.fr (195.83.24.193)  1.347 ms  1.188 ms  0.430 ms
 7 ksup.u-ga.fr (195.83.24.194)  0.189 ms  0.230 ms  0.198 ms
```

Notez que les adresses IP des routeurs et machines ci-dessus ont pu changer entre le moment où ce sujet a été rédigé et le moment où vous avez exécuté **traceroute**. Il est possible que votre résultat soit donc un peu différent.

Nous allons maintenant nous servir de Wireshark pour comprendre le fonctionnement de **traceroute**. 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 **traceroute** pour contacter **intranet.u-ga.fr**, mise en évidence sur la première ligne du résultat de **traceroute**. Entrez par exemple la chaîne de caractères : **(ip.addr eq 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.

18. À quelle machine **traceroute** envoie-t-il des messages ?
19. 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.
20. (Facultatif) Pour un **TTL** donné, combien de paquets sont émis ? En utilisant Wireshark, il arrive d'observer des paquets émis par **traceroute** avec un **TTL** supérieur au nombre de routeurs traversés (destination incluse) depuis votre machine. Qu'est ce que cela nous apprend sur le fonctionnement de **traceroute** ?
21. Faites un **traceroute** vers le site web de l'Université de Berkeley. **traceroute** parvient-il à révéler l'identité de tous les routeurs ?
22. Entre quels routeurs traverse-t-on l'Atlantique (regardez les temps intermédiaires) ?
23. Comment contraindre **traceroute** à n'afficher la route entre votre machine et le site web de l'Université de Berkeley qu'à partir du routeur numéro 10 par exemple ?
24. Faites un **traceroute** vers **slac.stanford.edu**. Que remarque-t-on ?
25. Pingez **slac.stanford.edu**. Cela fonctionne. Calculez le nombre de routeurs intermédiaires. Pourquoi **traceroute** ne marchait pas comme attendu ?
26. **mtr** est un outil de **traceroute** plus moderne que le vénérable **traceroute**. Essayez d'utiliser **mtr** vers **slac.stanford.edu** (appuyez ensuite sur la touche "q" pour quitter). Cela fonctionne-t-il ? Quelle différence avec **traceroute** pouvez-vous observer dans Wireshark ?

Pour les questions suivantes, on utilisera **mtr** en lieu et place de **traceroute**.

26. *Comparez l'itinéraire suivi jusqu'à l'université de Berlin (www.fu-berlin.de) d'une part, et jusqu'au site web de la ville de Berlin (www.berlin.de) d'autre part. Quelle raison principale peut justifier la différence d'itinéraire pour sortir de la France ?*

Pour la question suivante, utilisez un des serveurs de **traceroute** répartis dans le monde, accessibles par exemple en vous connectant au site web www.traceroute.org.

27. *Comparez le chemin aller et le chemin retour entre votre ordinateur et le site de l'Université de Stanford (www.slac.stanford.edu).*

3 Anatomie d'une Application Web (WWW)

Nous analysons maintenant la façon dont des applications utilisent le réseau. Le WWW, est selon Wikipedia :

The World Wide Web (WWW) is an open source information space where documents and other web resources are identified by URLs, interlinked by hypertext links, and can be accessed via the Internet. It has become known simply as the Web.

Ainsi, il définit trois éléments :

1. le nommage des objets (URL — Uniform Resource Locator) ;
2. le protocole de transfert d'objets (HTTP — HyperText Transfer Protocol) ;
3. un langage pour la spécification des documents (HTML — HyperText Markup Language).

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), accédez à <http://lig-membres.imag.fr/>.

28. *Bien que vous ayez demandé une autre page, le navigateur affiche <http://lig-membres.imag.fr/en/>. Avec WireShark, analysez les requêtes et les réponses HTTP, et expliquez pourquoi le navigateur a modifié l'URL.*

Dans les deux questions suivantes, reliez les observations visuelles sur la page, éventuellement celles venant du "source" HTML (**Ctrl-U** sous Firefox), et les échanges au niveau du protocole HTTP.

29. *Identifiez les liens et les images inclus dans cette page.*
30. *Quels sont les échanges au niveau du protocole HTTP ? (Utilisez le filtre Wireshark "**http**").*

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 d’accueil du site `duda.imag.fr` en utilisant l’outil `telnet` avec le port 80 :

```
telnet duda.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`, qui prend deux arguments séparés par un espace :

```
GET / HTTP/1.0
```

 (terminé par 2 retours à la ligne, donc avec une ligne vide).

Observez l’en-tête et le corps de la réponse.

30. *Quelle est la réponse du serveur si on utilise la méthode `HEAD` au lieu de `GET` ?*
31. *Quelle est la taille du contenu de la réponse ?*
32. *Quelle est la version du protocole utilisée par le serveur ?*
33. *Quelle est la différence entre l’en-tête de la requête engendrée par votre navigateur (Firefox) et celle que vous avez écrite ? N.B. : L’en-tête de la requête engendrée par votre navigateur est décodée par Wireshark.*

L’exemple ci-dessous détaille comment récupérer le contenu HTML de la page personnelle de l’un de vos encadrants de TP à l’aide de `telnet` qui est hébergée sur `lig-membres.imag.fr` en changeant la version de protocole utilisée (1.1).

Attention : pour utiliser HTTP/1.1, il faut donner le nom de host visé, sur une ligne qui suit la ligne du `GET` :

```
telnet lig-membres.imag.fr 80
...
GET /morine/ HTTP/1.1
Host: lig-membres.imag.fr
```

(N’oubliez pas les deux fois “entrée” !)

33. *Vérifiez que vous obtenez bien le code source de la page `http://lig-membres.imag.fr/morine/`.*
34. *Comparez les messages de connexion affichés sur le terminal lorsqu’on utilise `telnet` pour se connecter sur le port 80 de `duda.imag.fr` et de `lig-membres.imag.fr`. Que peut-on dire de ces deux adresses ?*

Pour confirmer ce que vous venez de conjecturer, on se propose de faire un telnet sur l’adresse IP de la question précédente :

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

avec la première fois l’hôte `duda.imag.fr` puis renouvelez la commande avec l’hôte `lig-membres.imag.fr`.

35. *Que pouvez vous dire de la version 1.1 d’HTTP ? Vous pouvez, pour vous aider, comparer avec les commandes en HTTP/1.0 sur la même adresse IP.*

Essayez maintenant la commande suivante :

```
telnet <ip_adress> 80
...
GET /morine/ HTTP/1.0
```

36. *Obtenez-vous le contenu de la page obtenue précédemment ? Pourquoi ?*

4 Première observation des couches réseau

4.1 Ethernet

À partir des trames réseaux obtenues dans Wireshark à la question 27, identifiez le premier segment TCP qui initie la connexion entre le client et le serveur HTTP.

37. *Relevez les adresses MAC source et destination de cette trame. À quels hôtes correspondent-elles ?*

4.2 IP

38. *Quelles sont les adresses source et destination du paquet IP ? À quels hôtes correspondent-elles ?*

39. *Repérez le champ Protocole dans l'en-tête. À quel type de Protocole correspond-t'il ?*

40. *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)*

4.3 TCP

Servez-vous de Wireshark pour observer les échanges au niveau du protocole TCP pendant le chargement d'une page Web. Essayez de trouver une page ne comportant pas trop d'images ni d'animations qui compliqueraient l'observation (comme par exemple ces exemples d'utilisation de gnuplot : http://gnuplot.sourceforge.net/demo_4.6/simple.html).

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

40. *Quelles connexions TCP sont impliquées dans le chargement de votre page Web ?*

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