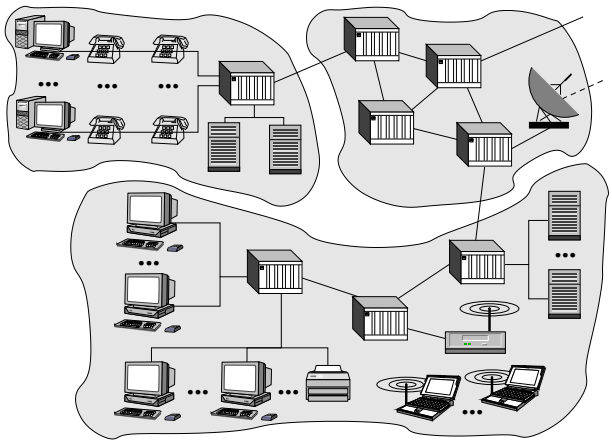
Master Informatique de l'Université Pierre & Marie Curie

ARES (MI011)

Architecture des Réseaux



Supports des **Labs n° 1, 2, 3, 4 et 5**
(version 6.3)

Olivier Fourmaux
(olivier.fourmaux@upmc.fr)

UPMC SORBONNE UNIVERSITÉS

ARES - Lab n°1

Introduction à la plateforme d'expérimentation

Ce premier support permet de se familiariser avec l'environnement d'expérimentation des Lab de l'ARES. Nous débuterons par quelques rappels sur l'analyse de trames (partie 1), puis nous présenterons l'outils de capture `wireshark` (partie 2). Nous détaillerons ensuite l'environnement pratique utilisé tout au long du semestre (la plateforme d'expérimentation de la spécialité RES du Master d'Informatique, partie 3). Nous présenterons les possibilités de capture de trafic réseau sur cette plateforme et terminerons par la réalisation d'un exercice pratique (partie 4). Avant la fin de la séance, n'oubliez pas de laisser l'environnement de travail dans son état initial (partie 5). Une annexe située à la fin de ce document est disponible pour vous aider dans vos analyses grâce à un rappel des diverses structures de données utilisées

1 Introduction à l'analyse de trames (sans ordinateur)

Pour étudier le trafic échangé dans un réseau, les administrateurs utilisent couramment des outils de capture matériels ou logiciels (appelés *sniffers*). Les outils logiciels reposent sur un équipement non dédié (un PC équipé d'une carte réseau) et un programme réalisant la capture et l'analyse multi-protocolaire (tel `tcpdump`, `wireshark` ou de nombreux autres logiciels).

1.1 Traces de trafic réseau

Les traces résultant de ces captures sont généralement réalisées au niveau de la couche liaison et consistent en une séquence de trames (potentiellement partielles). Les trames sont les recopies binaires (*binary dump*) de celles observées par la carte et sont structurées en octets, habituellement présentées en trois colonnes :

❶	❷	❸
0000	00 50 7f 05 7d 40 00 10 a4 86 2d 0b 08 00 45 00	.P..}@.. ..-...E.
0010	02 19 17 98 40 00 40 06 6c 14 0a 21 b6 b2 c0 37	....@.@. l..!...7
0020	34 28 84 b3 00 50 b6 94 b0 b8 24 67 89 e9 80 18	4(...P.. ..$g....
0030	16 d0 60 e4 00 00 01 01 08 0a 00 6f a7 32 00 00	..‘..... ...o.2..
0040	00 00 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31	..GET / HTTP/1.1
0050	.. ..	.. ..

❶ la première colonne indique, avec 4 chiffres hexadécimaux, le **rang** du premier octet de la ligne courante dans la trame ;

❷ la seconde affiche la **valeur hexadécimale** de 16 octets capturés à chaque ligne (un octet est représenté par deux caractères hexadécimaux) ;

❸ la dernière représente à chaque ligne les caractères ASCII correspondants aux 16 octets de la seconde colonne (la correspondance n'est significative que lorsque du texte "imprimable" se trouve encodé dans ces octets).

Quelques remarques importantes avant d'illustrer une analyse :

- Dans la suite, nous capturerons principalement des trames Ethernet. Les cartes réseau peuvent limiter les informations remontées au noyau, ainsi la représentation des trames ne comporte **ni préambule, ni CRC.**
- Dans le monde professionnel, vous devrez respecter les usages afin de communiquer efficacement. Ainsi, respectez **impérativement** les conventions d'écriture adaptées aux différents champs que vous analysez, par exemple :
  - **Adresses Ethernet** : hexadécimale double pointée (ex : `00:50:04:ef:6b:18`)
  - **Type Ethernet** : hexadécimale (ex : `0x0806`)
  - **Adresses IP** : décimale pointée (ex : `10.1.1.3`)
  - **Numéro de protocole** et **numéro de port** : décimale (ex : 17)
  - ...

1.2 Analyse manuelle

Afin de bien intégrer les mécanismes mis en oeuvre par un outil d'analyse, étudions **sur papier** le début d'une trame capturée sur un réseau Ethernet. Cet exercice fastidieux est nécessaire pour acquérir une bonne compréhension des mécanismes d'encapsulation et se prémunir des potentielles erreurs d'interprétation des outils automatisés. Les structures des protocoles rencontrés sont rappelées **page 13** :

```
0000  00 50 7f 05 7d 40 00 10  a4 86 2d 0b 08 00 45 00   .P..}@.. ..-...E.
0010  02 19 17 98 40 00 40 06  6c 14 0a 21 b6 b2 c0 37   ....@.@. l..!...7
0020  34 28 84 b3 00 50 b6 94  b0 b8 24 67 89 e9 80 18   4(...P.. ..$g....
0030  16 d0 60 e4 00 00 01 01  08 0a 00 6f a7 32 00 00   ..‘..... ...o.2..
0040  00 00 47 45 54 20 2f 20  48 54 54 50 2f 31 2e 31   ..GET /  HTTP/1.1
0050  0d 0a 48 6f 73 74 3a 20  77 77 77 2e 78 69 72 63   ..Host:  www.xirc
0060  6f 6d 2e 63 6f 6d 0d 0a  55 73 65 72 2d 41 67 65   om.com.. User-Age
0070  6e 74 3a 20 4d 6f 7a 69  6c 6c 61 2f 35 2e 30 20   nt: Mozi lla/5.0
0080  28 58 31 31 3b 20 55 3b  20 4c 69 6e 75 78 20 69   (X11; U;  Linux i
0090  36 38 36 3b 20 65 6e 2d  55 53 3b 20 72 76 3a 31   686; en- US; rv:1
00a0  2e 30 2e 30 29 20 47 65  63 6b 6f 2f 32 30 30 32   .0.0) Ge cko/2002
00b0  30 36 32 33 20 44 65 62  69 61 6e 2f 31 2e 30 2e   0623 Deb ian/1.0.
00c0  .. ..
```

1. Détaillez la structure de la **trame** en dessinant directement ses délimitations sur la trace à analyser.
2. Quelles informations de la couche liaison pouvez-vous observer ?

3. Représentez la structure du paquet directement sur la trace à analyser. Quelle est la taille de ce paquet et qu'en déduisez-vous ? Le paquet contient-il des options et quel en est l'effet sur la structure du paquet ? Précisez la source et le destinataire du paquet.

4. Représentez la structure des données transportées par le paquet directement sur la trace. Quel est le protocole de transport utilisé ? Quels sont les ports utilisés ? Quelle est leur signification ?

5. *Il n'y a pas de documentation correspondant à la couche application à la fin du document,* malgré cela, pouvez vous observer des informations associées à ce niveau dans la trace ?

2 Analyse de trames avec `wireshark`

Le logiciel `wireshark`[1] est outils de capture de trame et d'analyse de protocoles. Celui-ci peut utiliser directement l'interface de votre machine pour capturer des trames circulant sur le réseau local puis les analyser. Pour cette section, nous allons nous limiter à la fonction d'**analyse de protocole** en chargeant une capture déjà réalisée à partir d'un fichier.

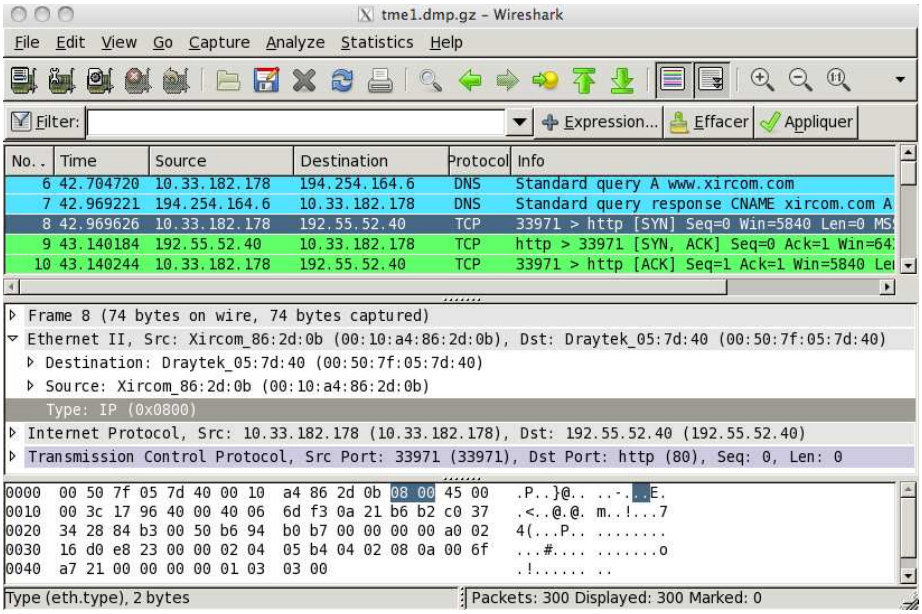


FIGURE 1 – Fenêtre principale de `wireshark`

Pour pouvoir travailler les exercices à l'extérieur de l'université, vous pouvez recopier les traces réalisées pendants les séances ou télécharger celles disponibles dans la page web suivante :
`http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`

2.1 Introduction à `wireshark`

Sur la machine face à votre binôme, connectez-vous à votre compte sous **GNU/Linux**. Cet ordinateur, géré par l'ARI[2], accède à l'Internet et à diverses ressources sensibles, vous n'y avez pas les droits d'administrateur *(root)*. Or les logiciels d'analyse de trames requièrent ceux-ci pour réaliser des captures à partir de votre carte réseau. Pour utiliser seulement la partie analyse multi-protocolaire de ces logiciels, vous pouvez les exécuter avec les droits limités des utilisateurs, ce qu'il faudra donc veiller à faire explicitement sur les machines de l'ARI. Dans les sections suivantes nous étudierons comment réaliser des captures, mais sur d'autres machines.

[1] `wireshark` est un logiciel libre. Il est disponible sur un grand nombre de plates-formes matérielles et systèmes d'exploitation (outre les machines à architecture **i686** avec système **GNU/Linux** que vous utilisez actuellement). Vous pouvez le télécharger sur `http://www.wireshark.org`.
[2] Atelier de Ressources Informatiques dédié aux enseignements du domaine Informatique de la faculté d'Ingénierie de l'UPMC.

Recherchez dans les sous-menu du menu "Application", vous devez trouvez un item "Wireshark". Sélectionnez le et exécuter le sans les droits d'administrateur[3].

Une fois l'application lancée, la nouvelle fenêtre apparue est initialement vide car aucune capture n'a été réalisée ou chargée. Une barre de menu se trouve en haut de celle-ci. Pour charger une trace à étudier, cliquez sur le menu "File" et sélectionnez "Open". Une fenêtre de sélection de fichier "Open Capture Files" apparaît. Choisissez le fichier :

`/Infos/lmd/2013/master/ue/ares-2013oct/tme1.dmp.gz`

Ne pas spécifier de filtres dans le champ "Filter" (nous y reviendrons plus loin). Désactivez : ❑ "Enable MAC name resolution", ❑ "Enable network name resolution" et ❑ "Enable transport name resolution". Validez avec Ouvrir : La trace d'une capture précédemment réalisée est chargée et vous allez pouvoir l'analyser. Vous devez observer dans la fenêtre de l'application un affichage similaire à celui présenté dans la FIGURE 1.

1. Décrivez le contenu des trois fenêtres proposées par `wireshark`.
2. Dans quels formats sont représentés les données de la troisième fenêtre ?
3. Quels sont les différents protocoles que vous pouvez observer dans la capture affichée ?
4. Combien de protocoles est capable d'analyser la version de `wireshark` que vous utilisez ?

2.2 Filtres d'affichage et de coloriage de `wireshark`

1. Avec la rubrique d'aide (cliquez sur le menu "Help" et sélectionnez "Manual Pages"), décrivez la syntaxe utilisée par les filtres d'affichage et de coloriage *(Display filters)*. Ces filtres ne doivent pas être confondus avec les filtres de capture qui répondent à une autre syntaxe que nous n'utiliserons pas.
2. Décrivez un filtre qui ne sélectionne que les trames contenant le protocole applicatif NTP. Pour vous aider, le menu "Analyse" propose "Display Filters..." qui affiche une fenêtre d'édition de filtre. Le bouton +Expression autorise à la création interactive de l'expression correspondante. Appliquez ce filtre. Qu'observez-vous ?
3. Supprimez le filtre précédent et coloriez en violet les trames contenant du protocole NTP.
4. Vous pouvez également combiner les filtres à l'aide des opérateurs booléen usuels. Filtrez l'affichage pour ne conserver que les trames contenant du protocole NTP et celles contenant du protocole DNS.

2.3 Analyse d'un trafic HTTP

Dans la continuité de la trame étudiée manuellement dans la section précédente :

1. Pouvez-vous retrouver la trame analysée manuellement dans la trace que vous avez chargée ? Le cas échéant, confrontez votre analyse à celle réalisée par `wireshark`.
2. Sélectionnez et affichez **toutes** les trames relatives à la connexion TCP démarrant à la trame 8, puis coloriez en rouge seulement celles contenant des données HTTP.
3. Décrivez ce que vous observez dans le reste de la trace. Précisez s'il y a plusieurs connexions, et le cas échéant, leur relation.
4. Peut-on visualiser simplement le contenu applicatif d'une connexion TCP avec `wireshark` ?

3 Présentation de la plateforme d'experimentation

La principale limitation des postes ARI, sur lesquels vous travaillez, est l'impossibilité de réaliser des captures par vous-mêmes en temps réel. Afin de pallier cette limitation et d'offrir l'accès à un grand nombre d'équipements réseau, une plateforme d'expérimentation a été installée dans la salle.

L'ajout de cette plateforme réseau permet de faire évoluer le poste ARI habituel d'un simple PC vers un poste d'accès et de contrôle des différents élément de la plateforme (machines terminales, commutateurs et routeurs), tout en y conservant les services usuels de l'ARI (accès aux comptes utilisateurs, aux logiciels habituels et à l'Internet). Cette évolution est présentée dans la FIGURE 2.

[3] Si le choix d'une exécution avec ou sans les droits d'administrateur n'apparait pas, vous aurez peut être à spécifier ultérieurement. En cas d'échec, généralement lié à l'exécution proposée par votre environnement qui essaye d'utiliser le mode administrateur (mode par défaut de la commande `wireshark` relative au `$PATH` local), vous pouvez démarrer en mode textuel (dans un terminal) : Tapez alors la commande `/usr/sbin/wireshark`.

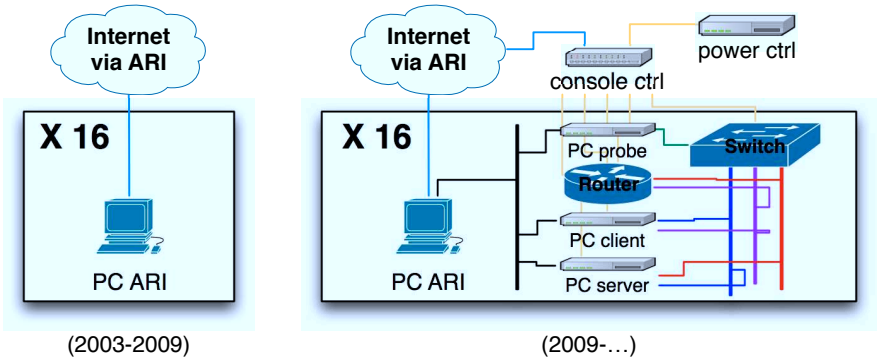
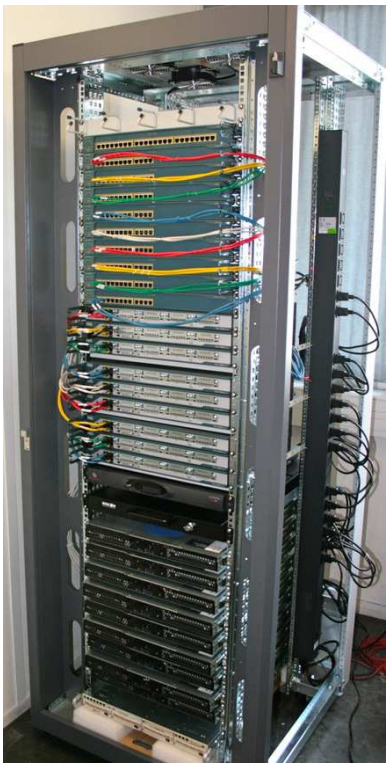


FIGURE 2 – Evolution du poste étudiant

3.1 Composition matérielle de la plateforme d'expérimentation

Deux baies *(racks)* 19" hautes de 42U concentrent les équipements des plates-formes de la salle :



- **des commutateurs (switchs) CISCO Catalyst 2950-12**, 12 ports 100BaseT :
  - accès aux fonctions de contrôle des ports et VLAN
  - gestion de la recopie de port (pour la capture de trames)
- **des routeurs CISCO 2801**, 2 ports 10/100BaseT, IOS 12.4 avec deux niveaux de service :
  - **IP Base :** IPv4, RIP, OSPF, IGMP, Netflows, QoS, RSVP, DiffServ, DHCP, NAT, SNMP, RMON, NTP, L2TP, AAA...
  - **Advanced IP Services :** IOS IP Base + IPv6, BGP, Mobile IP, VoIP, SIP, H323, Firewall, IPSEC, VPN, AES...
- **des PC "rackables" 1U**, mono-proc. Intel XEON QuadCore (X3220, 2.4GHz), 4Go RAM DDR2 667MHz, HD 250Go 7K2Tm, 4 NIC Ethernet 1000BaseT, exécutant des machines virtuelles (VM) avec :
  - **Debian GNU/Linux** incluant un environnement réseau Unix classique (Telnet, SSH, FTP, TFTP, SCP, SFTP, HTTP, SMTP, POP, IMAP, Webmail, SNMP, DNS...)

3.2 Usage étudiant de la plateforme d'expérimentation

Chaque binôme étudiant accède à la plateforme via un poste générique de l'ARI de la salle. Ces postes sont des PC équipés de deux cartes réseau. L'une permet l'accès au réseau habituel de l'ARI et donc à l'Internet, l'autre permet l'accès direct aux équipements de la plateforme. Ainsi, l'accès à la plateforme se fait soit physiquement via le poste ARI de cette salle (31-208) ou à distance via SSH sur ce même poste (`ssh -X ari-31-208-`**N**). Il n'y a pas de routage ou relayage entre le réseau de l'ARI et celui de la plateforme assurant ainsi l'isolation du réseau expérimentation.

Les postes `ari-31-208-01` à `ari-31-208-08` sont connectés sur la baie 1 et les postes `ari-31-208-09` à `ari-31-208-16` sur la baie 2.

Appelons **N** la valeur du dernier nombre du nom de la machine ARI utilisée. Le poste ARI **N** peut accéder directement à 3 équipements dédiés de la plateforme d'expérimentation :

- le commutateur **N**
- le routeur **N**
- le PC **N** utilisé pour faire tourner plusieurs machines virtuelles (VM) :
  - la VM "client" **N1**
  - la VM "sonde" **N2**
  - la VM "serveur" **N3**

La configuration des VM est présentée dans la FIGURE 3.

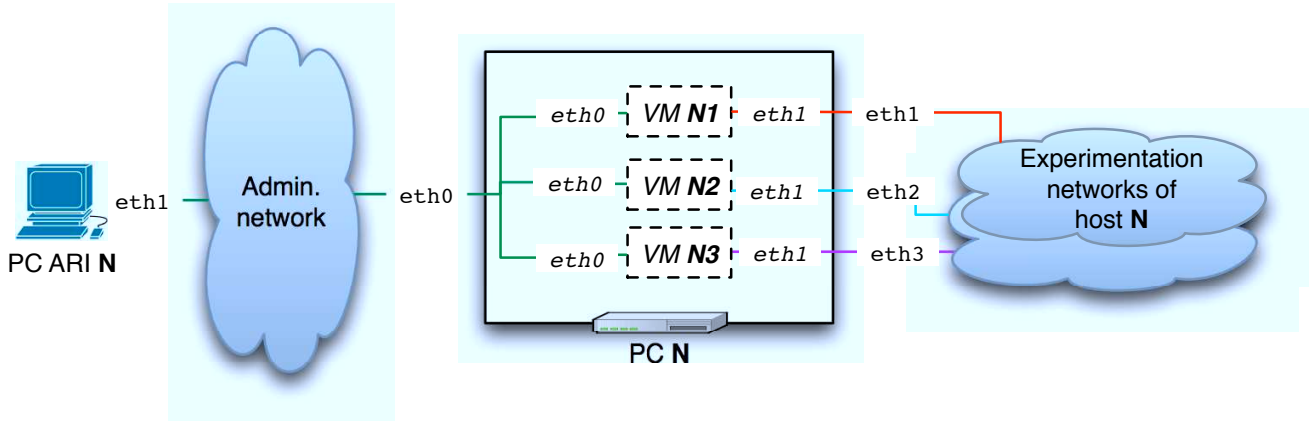


FIGURE 3 – Configuration des 3 VM de la plate-forme

Les identifiants et mots de passe nécessaires des différents équipements seront fournis lors des séances par les encadrants selon les besoins.

3.3 Topologies réalisables

La plateforme d'expérimentation a pour but de proposer différentes topologies réseau virtuelles à partir d'une configuration physique figée (les baies sont fermées et inaccessibles aux étudiants). La topologie physique correspond donc au câblage reliant le poste ARI aux équipements directement accessibles à celui-ci. La FIGURE 4 présente ces liens physiques sur lesquels transiteront vos paquets.

3.3.1 Topologie 1 (sans routeur – premiers labs)

A partir de la topologie physique, une première configuration virtuelle correspond à un simple réseau local sur lequel s'échange du trafic entre deux hôtes. La FIGURE 5 représente le poste de l'ARI relié aux équipements étudiés (VM "client", VM "sonde", VM "serveur" et commutateurs) via un réseau d'administration (VLAN 200). Une fois connecté aux VM de la plateforme, les applications client/serveur peuvent être lancées pour échanger du trafic sur un réseau dédié (VLAN **N**1) et la VM "sonde" peut capturer celui-ci avec une application d'analyse de trames.

3.3.2 Topologie 2 (avec un routeur – labs suivants)

La seconde configuration virtuelle intègre un routeur entre deux réseaux locaux avec un hôte sur chacun. Elle est présentée sur la FIGURE 6. Le réseau d'administration (VLAN 200) est toujours présent pour accéder aux équipements (dont le routeur). La modification porte principalement sur les deux réseaux dédiés au trafic d'expérimentation de chaque coté du routeur (VLAN **N**1 et VLAN **N**2). Cette configuration permettra d'étudier le comportement du trafic routé, grâce à la VM "sonde" qui peut capturer celui-ci avec une application d'analyse de trames.

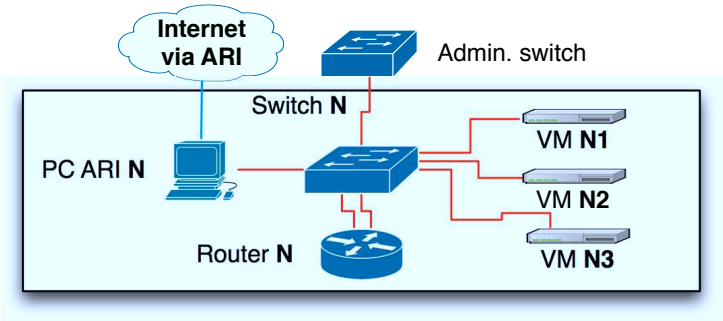


FIGURE 4 – Topologie physique associée à un poste ARI

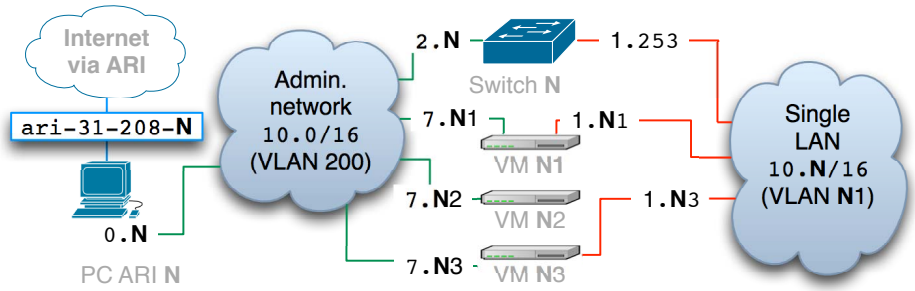


FIGURE 5 – Topologie virtuelle 1 (un LAN)

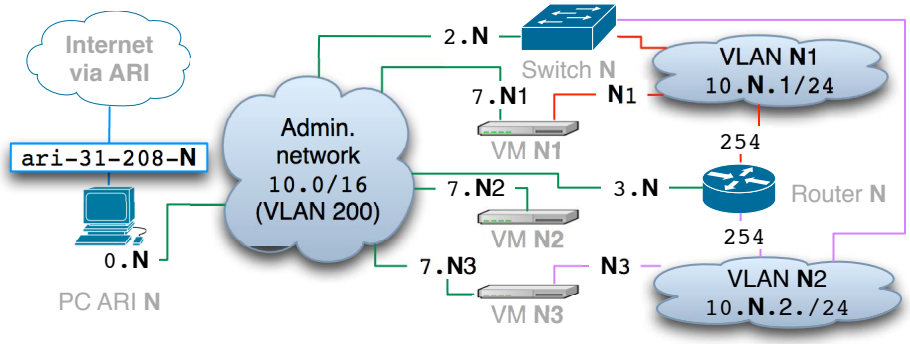


FIGURE 6 – Topologie virtuelle 2 (deux LAN et un routeur)

3.3.3 Topologie 3 (avec plusieurs routeurs – dernier lab et pour d'autres cours)

Des configurations plus évoluées seront également proposées, en particulier pour aborder le routage multi-saut (plusieurs routeurs avec les protocoles RIP, OSPF ou BGP) et servir à d'autres U.E. de la spécialité RES.

3.3.4 Conventions d'adressages IPv4 relative au poste N

Deux types de VLAN sont utilisés sur la plate-forme d'expérimentation :

- Le VLAN d'administration (VLAN 200) et ses adresses IPv4 d'administration pour :
  - l'interface `eth1` du poste ARI **N** (accès à la plate-forme via le réseau d'administration) : `10.0.0.`**N**
  - les commutateurs : `10.0.2.`**N**
  - les routeurs : `10.0.3.`**N**
  - les VM "client" **N1** (`eth0`) : `10.0.7.`**N**1
  - les VM "sonde" **N2** (`eth0`) : `10.0.7.`**N**2
  - les VM "serveur" **N3** (`eth0`) : `10.0.7.`**N**3
- Les VLAN d'expérimentation (VLAN **N**v avec $0 < v$) et leurs adresses IPv4 d'expérimentation pour :
  - les VM "client" **N**1 des VLAN **N**v (`eth1`) : `10.`**N**`.`v`.`**N**1
  - les VM "server" **N**3 des VLAN **N**v (`eth1`) : `10.`**N**`.`v`.`**N**3
  - les routeurs des VLAN **N**v : `10.`**N**`.`v`.254`

3.4 Utilisation courante pour générer du trafic et le capturer

Le poste ARI est relié au réseau d'administration de la plate-forme d'expérimentation par son interface `eth1` (voir sur la FIGURE 3). La commande Unix `/sbin/ifconfig` permet de vérifier la configuration des interfaces d'une machine (vous pouvez préciser le nom de l'interface à la suite). Si l'interface `eth1` est inexistante, redémarrez votre poste ARI.

3.4.1 Contrôle à distance des 3 VM de la plateforme

Quelle que soit la topologie utilisée, pour démarrer toute utilisation de la plateforme, il est nécessaire de contrôler les hôtes requis via des sessions SSH depuis le poste de l'ARI sur lequel vous travaillez. Par exemple, si nous souhaitons utiliser les trois VM de la plateforme qui nous sont associés, il nous faudra ouvrir trois terminaux textuels à travers lesquels les équipements concernés seront supervisés (le login est `etudiant`, ou `root` si besoin, et le mot de passe sera fourni par votre encadrant). En travaillant à partir du poste **N** :

- fenêtre 1 (hôte "client") : tapez `ssh -X etudiant@10.0.7.`**N**1
- fenêtre 2 (hôte "sonde") : tapez `ssh -X root@10.0.7.`**N**2
- fenêtre 3 (hôte "serveur") : tapez `ssh -X etudiant@10.0.7.`**N**3

L'option "`-X`" signifie que l'environnement graphique de la machine distante (fenêtres X11) sera redirigé sur le poste local. Cela n'est pas nécessaire si le contrôle est uniquement textuel.

Les VM de la plate-forme sont reliées au réseau d'administration par leur interface `eth0` et aux réseaux d'expérimentation via `eth1` (voir sur la FIGURE 3). La commande Unix `/sbin/ifconfig` permet de vérifier la configuration des interfaces. Dans les 3 terminaux précédemment lancés, exécutez cette commande. Vous devez observer un affichage similaire à celui de la FIGURE 7.

3.4.2 Exécution de `wireshark` et lancement de la capture

Dans la suite, nous étudierons principalement des applications et des protocoles client/serveur. La VM "client" et la VM "serveur" seront donc utilisées pour analyser les échanges réseau associés. La VM "sonde" va permettre de faire des captures de trafic à l'aide du logiciel `wireshark`. Celui-ci pourra utiliser directement l'interface `eth1` de cette machine pour écouter les informations circulant sur le réseau d'expérimentation. Cette interface doit être configurée en mode "*promiscuous*" afin d'accéder à tout le trafic et pas seulement celui qui lui est est explicitement destiné. Pour utiliser ce mode, l'application doit être exécutée avec les privilèges de l'administrateur (utilisateur `root` sous Unix). C'est pour cela que l'on se connecte à la "sonde" avec cet

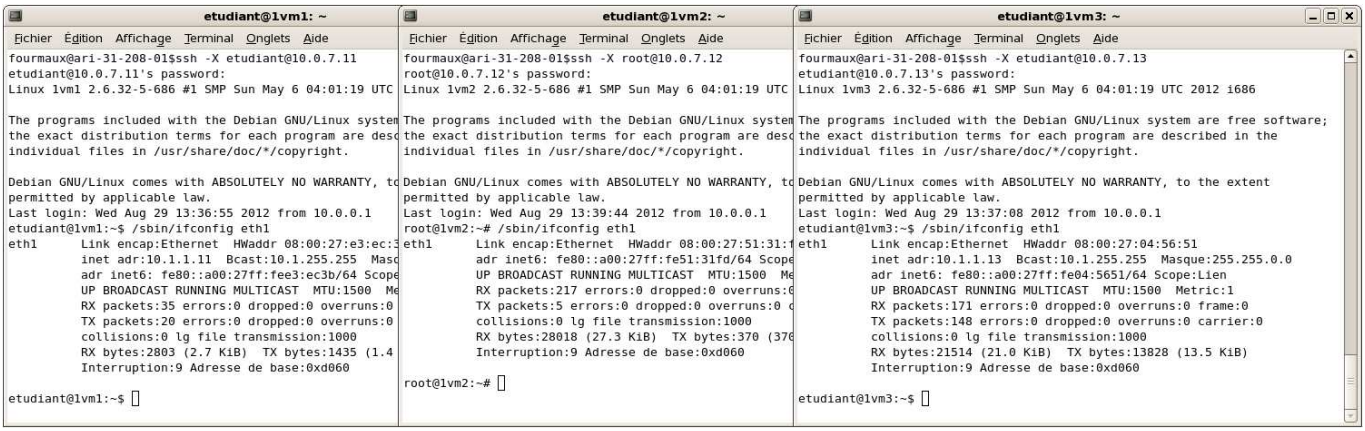


FIGURE 7 – Sessions SSH à partir du poste `ari-31-208-01`

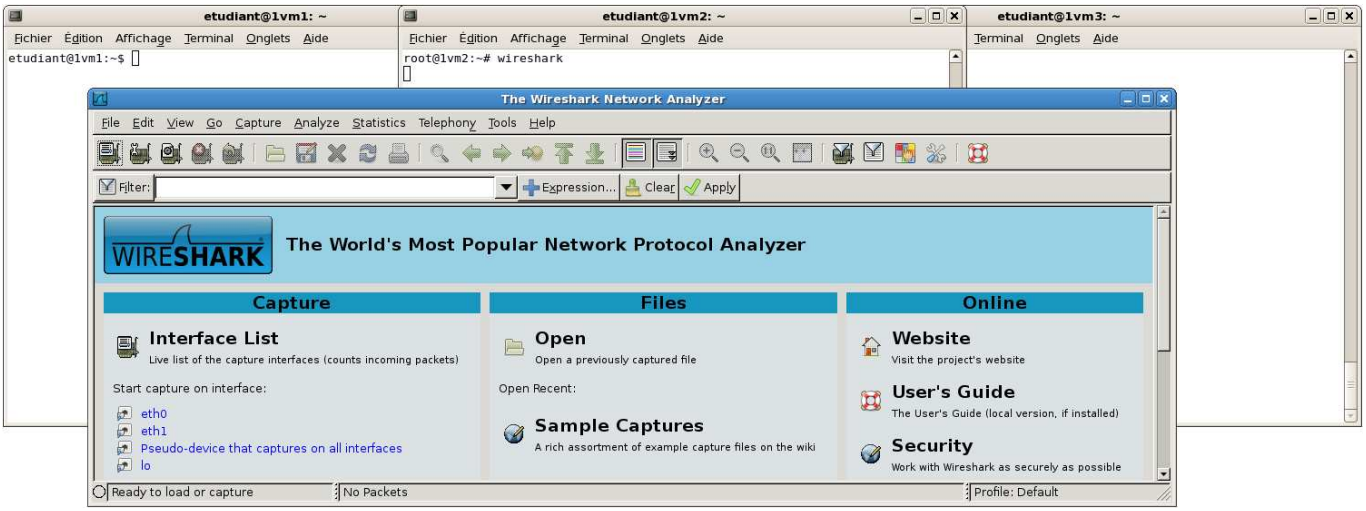


FIGURE 8 – Démarrage de `wireshark` avec les privilèges administrateur

utilisateur . **Attention, l'utilisation de commandes en tant qu'administrateur peut sérieusement impacter le système que vous utilisez. Soyez donc précautionneux et effectuez impérativement des sauvegardes de tout les fichiers que vous modifiez.**

Une fois que vous êtes l'utilisateur `root`, lancez `wireshark` de la fenêtre de la "sonde" et une nouvelle fenêtre en provenance de la VM "sonde" apparaîtra sur votre écran (grâce à la redirection X11). L'affichage doit être similaire à la FIGURE 8. Cliquez sur le menu "Capture" et sélectionnez "Interfaces...". Une fenêtre présentant les différentes interfaces de la machine apparaît (voir la FIGURE 9). Sélectionnez le champ "Options" de l'interface `eth1` (elle n'a pas d'adresse IPv4, seulement une adresse IPv6).

Une nouvelle fenêtre apparaît (voir la FIGURE 10). Ne pas spécifier de filtres dans le champ "Capture Filter". Désactivez :

- ❑ "Enable MAC name resolution"
- ❑ "Enable network name resolution"
- ❑ "Enable transport name resolution"

Initiez la capture avec Start : La capture démarre et vous pouvez observer du trafic en générant, par exemple, des demandes d'écho de la VM "client" vers la VM "serveur". Utilisez pour cela la commande Unix `ping 10.N.1.N3` dans la fenêtre "client", puis observez la capture dans la fenêtre de `wireshark` (voir la FIGURE 11).

Pour arrêter le `ping`, tapez `Ctrl-C` dans la fenêtre de la VM "client". N'oubliez pas d'arrêter la capture avec le bouton Stop dans la fenêtre de capture.

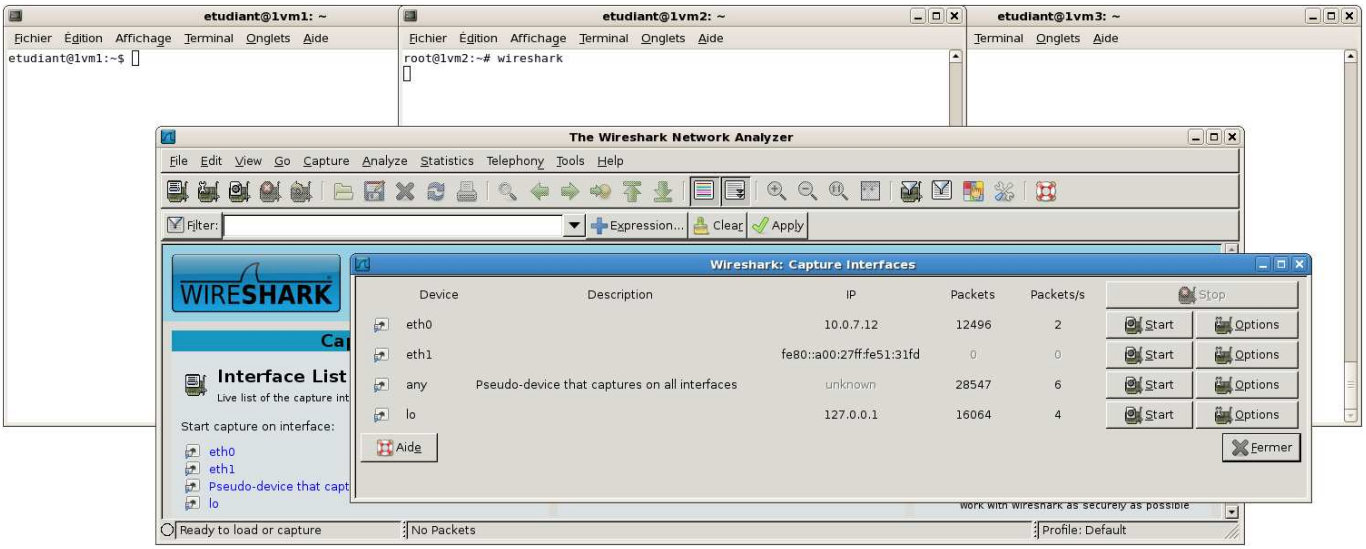


FIGURE 9 – Démarrage d'une capture à partir de la VM "sonde" : liste des interfaces

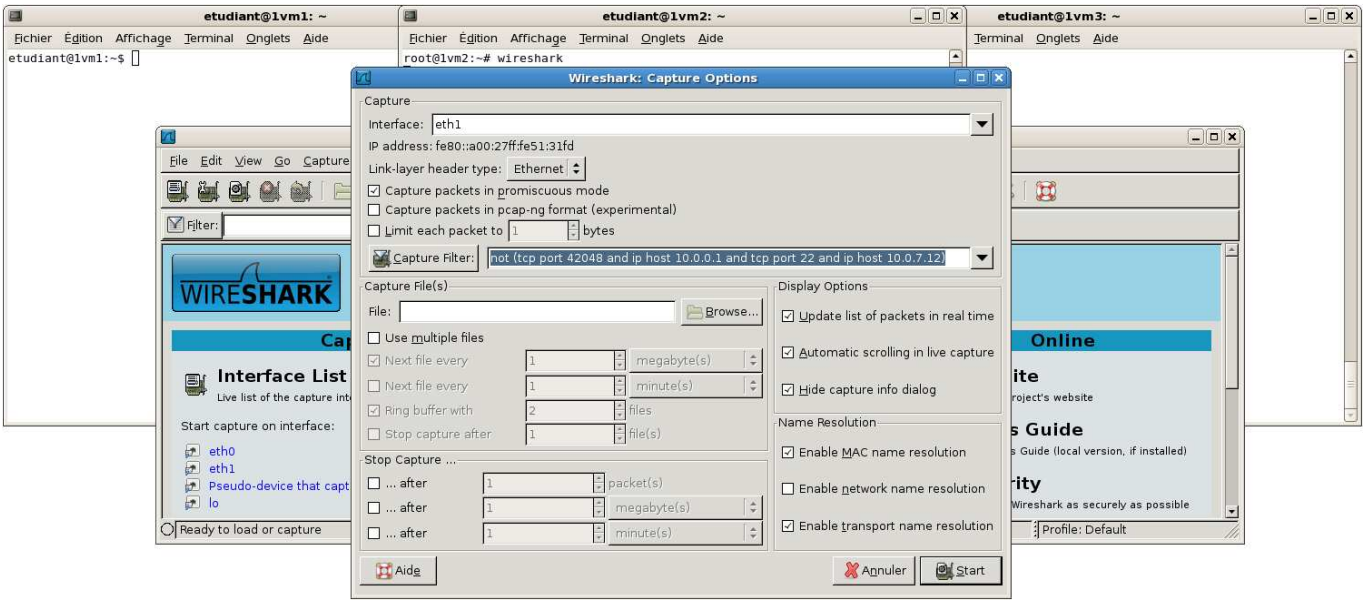


FIGURE 10 – Options de capture associées à l'interface `eth1`

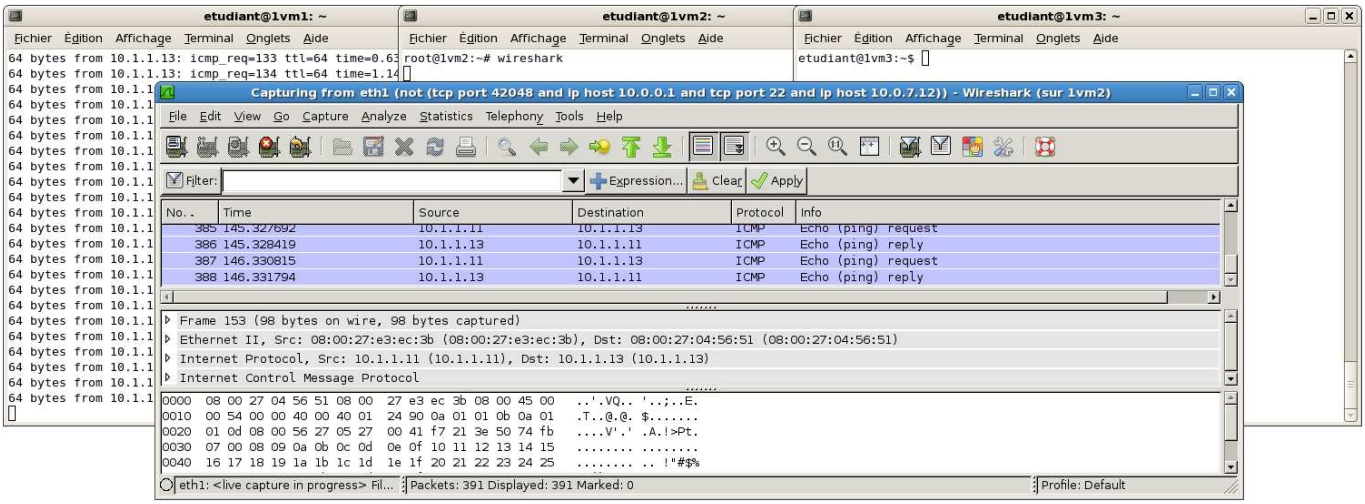


FIGURE 11 – Capture de trafic `ping` (paquets ICMP)

4 Exemple de capture et analyse de trames sur la plateforme réseau

On se place dans la première topologie virtuelle (un simple réseau local sur lequel s'échange du trafic entre deux hôtes directement connectés). La configurations des différents équipements est déjà en place pour cette séance. Le poste ARI permet de se connecter directement aux équipements nécéssaires (VM "client", VM "sonde", VM "serveur" et commutateur) via un LAN d'administration (VLAN 200). Le navigateur `firefox` (sur la VM "clients") et le serveur web `apache` (sur la VM "serveur") peuvent échanger du trafic sur un LAN dédié (VLAN **N**1). La VM "sonde" peut capturer celui-ci avec `wireshark`.

4.1 Capture d'un trafic HTTP

En travaillant à partir du poste **N** :

- Se connecter sur les 3 hôtes de la plateforme (si ce n'est déjà fait), avec le login `etudiant` et le mot de passe fourni par votre encadrant.
  - fenêtre 1 (hôte "client") : tapez `ssh -X etudiant@10.0.7.`**N**1
  - fenêtre 2 (hôte "sonde") : tapez `ssh -X root@10.0.7.`**N**2
  - fenêtre 3 (hôte "serveur") : tapez `ssh -X etudiant@10.0.7.`**N**3
- Vérifiez que le serveur HTTP tourne sur `10.0.7.`**N**3 (fenêtre 3)
  - recherchez le processus du serveur web, tapez `ps aux | grep apache`
  - visualisez la configuration des interfaces pour vérifier l'adresse IP du serveur (`/sbin/ifconfig eth1`)
- Lancez l'analyseur sur `10.0.7.`**N**2 (fenêtre 2)
  - lancez l'analyseur, tapez : `wireshark`
  - initier la capture sur l'interface `eth1`, comme indiqué précédemment
- Démarrez un client web sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client, tapez : `firefox` (la version de ce navigateur sur Debian s'appelle `iceweasel`)
  - ouvrez dans le navigateur la page `http://10.`**N**`.1.`**N**3
- Observez la capture dans la fenêtre de `wireshark`, vous devez voir s'afficher quelque chose de similaire à la FIGURE 12. N'oubliez pas de terminer la capture.

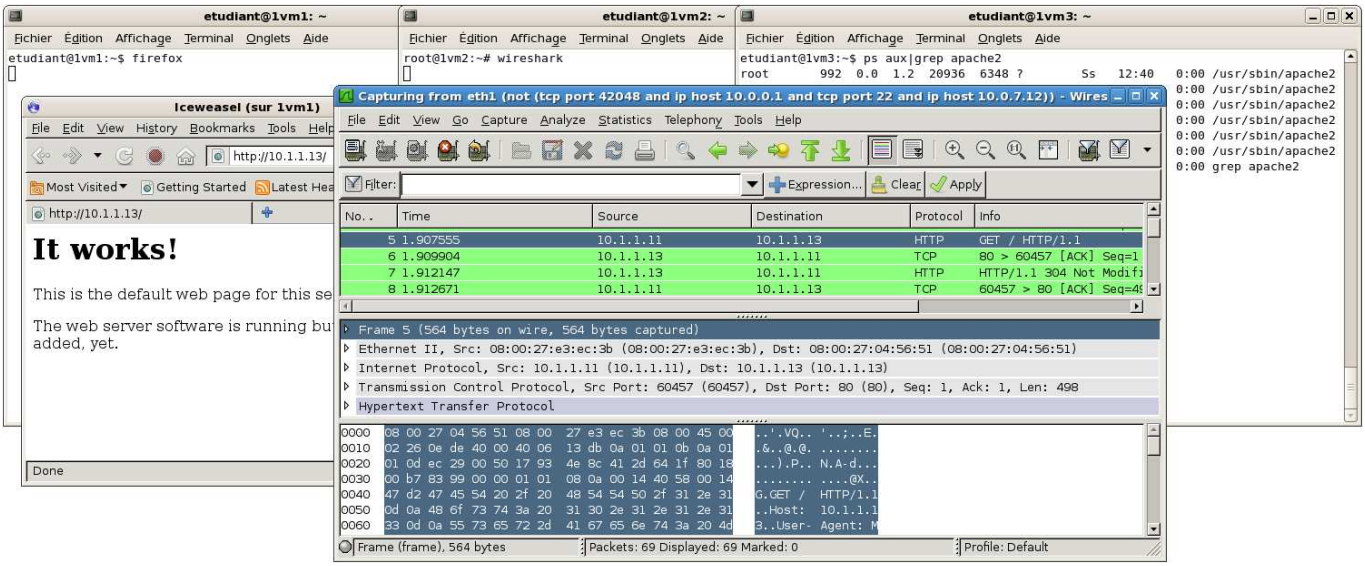


FIGURE 12 – Capture de trafic HTTP

4.2 Analyse du trafic HTTP capturé

Avec la trace réalisée précédemment :

1. Sélectionnez **toutes** les trames contenant des données HTTP.
2. Décrivez ce que vous observez, et s'il y a plusieurs connexions, quelle est leur relation ?
3. Observez le code source de la page affichée par le navigateur sur le client. Essayez de retrouver où se trouve cette page sur le serveur et vérifiez que c'est ce contenu qui est passé sur le réseau.

5 Avant de quitter la salle

- Avant de terminer vos connexions sur les équipements de la plateforme, supprimez vos fichiers et modifications effectuées afin de retrouver l'état initial.

Structure de la trame Ethernet

```
.......+-48bits--+-48bits--+16b-+- - - - -+......
.(Pré.)| adresse | adresse |type| données |(CRC).
.      | dest.   | source  |    |         |     .
.......+---------+---------+----+- - - - -+......

Quelques types : 0x0200 = XEROX PUP
                 0x0800 = DoD Internet (IPv4)
                 0x0806 = ARP
                 0x8035 = RARP
```

Structure ARP

```
+16b-+-16b-+8b+8b+16b+--lgHW--+lgP-+--lgHW--+lgP-+
|type|type |lg|lg|Op |Emetteur|Emt.|Récept. |Rcpt|
|HW  |Proto|HW|P |   |adr. HW |adrP|adr. HW |adrP|
+----+-----+--+--+---+--------+----+--------+----+

Quelques types : 0x0001 = Ethernet
                 0x0800 = DoD Internet (IPv4)
Opérations : 0x0001 = Requête
             0x0002 = Réponse
```

Structure du paquet IPv4

```
<-----------------32bits----------------------->
<-4b->      <--8bits--->|<--------16bits-------->
+-----+-----+-----------+----------------------+
| Ver | IHL | TOS       | Longueur totale (octet)
+-----+-----+-----------+--+-------------------+
| Identificateur        |Fl| FO                |
+-----------+-----------+--+-------------------+
| TTL       | Protocole | Somme de ctrl (entête)|
+-----------+-----------+----------------------+
| Adresse Source                               |
+----------------------------------------------+
| Adresse Destination                          |
+----------------------------------------------+
...     Options
+----------------------------------------------+
...     Données
+----------------------------------------------+

Ver = Version d'IP
IHL = Longueur de l'en-tête IP (en mots de 32 bits)
TOS = Type de service (zero généralement)
Fl (3 premiers bits) = Bits pour la fragmentation
 * 1er = Reservé
 * 2me = Ne pas fragmenter
 * 3me = Fragment suivant existe
FO (13 bits suivants) = Décalage du fragment
TTL = Durée de vie restante
Quelques protocoles:              8 = EGP
        1 = ICMP                 11 = GLOUP
        4 = IP (encapsulation)   17 = UDP
        6 = TCP                  46 = RSVP
```

Structure du paquet ICMP

```
<-----------------32bits----------------------->
+-----------+-----------+----------------------+
|  Type     |  Code     | Somme de contrôle (msg)
+-----------+-----------+----------------------+
|  Variable (généralement non utilisé)         |
+----------------------------------------------+
...     Datagramme original + 8 octets
+----------------------------------------------+

Quelques types ICMP :  8 = Demande d'écho
                       0 = Réponse d'echo
                      11 = Durée de vie écoulée
                      12 = Erreur de paramètre
```

Structure de segment TCP

```
<-----------------32bits----------------------->
<-4b->         <-6bits-><--------16bits-------->
+----------------------+----------------------+
| Port Source          |  Port Destination    |
+----------------------+----------------------+
| Numéro de Séquence                          |
+---------------------------------------------+
| Numéro d'Acquittement                       |
+-----+-------+---------+--------------------+
| THL |       |  Flag   |  Taille Fenêtre    |
+-----+-------+---------+--------------------+
| Somme de ctrl (message)  Pointeur d'Urgence |
+----------------------+----------------------+
...      Options
+---------------------------------------------+
...      Données
+---------------------------------------------+

THL = Longueur de l'entête TCP sur 4 bits (*32bits)
Flags = indicateur codé sur 6 bits gauche à droite
 * 1er = Données urgentes (URG)
 * 2me = Acquittement (ACK)
 * 3me = Données immédiates (PSH)
 * 4me = Réinitialisation (RST)
 * 5me = Synchronisation (SYN)
 * 6me = Terminaison (FIN)
Options = suites d'option codées sur
 * 1 octet à 00 = Fin des options
 * 1 octet à 01 = NOP (pas d'opération)
 * plusieurs octets de type TLV
   T = un octet de type:
     2 Annonce de la taille max. du segment
     3 Adaptation de la taille de la fenêtre
     4 Autorisation des acquittements sélectifs
     8 Estampilles temporelles
   L = un octet pour la taille totale de l'option
   V = valeur de l'option (sur L-2 octets)
```

Structure de datagramme UDP

```
<-----------------32bits----------------------->
+----------------------+----------------------+
| Port Source          | Port Destination     |
+----------------------+----------------------+
| Longueur UDP         | Somme de ctrl (message)
+----------------------+----------------------+
...      Données
+---------------------------------------------+
```

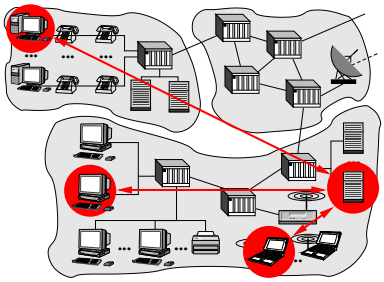
Services associés aux ports

ftp-data	20/tcp		
ftp	21/tcp		
ssh	22/tcp	ssh	22/udp
telnet	23/tcp		
smtp	25/tcp		
domain	53/tcp	domain	53/udp
		tftp	69/udp
www	80/tcp	www	80/udp
kerberos	88/tcp	kerberos	88/udp
pop-3	110/tcp	pop-3	110/udp
		snmp	161/udp
		snmp-trap	162/udp

ARES - Lab n°2

Couche application (1) : Telnet, SSH, FTP, TFTP et Web

Lors du Lab n°1, vous avez appris comment utiliser la plateforme d'expérimentation et vous l'avez exploitée afin de générer et d'analyser des traces assez simples de la couche application, contenant du trafic web. Pour le Lab n°2, vous allez explorer la couche application beaucoup plus en détail, en étudiant les protocoles suivants : TELNET, RLOGIN, SSH, FTP, SFTP, TFTP et HTTP. Pour chacun, vous allez générer du trafic réel que vous allez capturer et analyser avec l'outil `wireshark`. Vous utiliserez également le RFC de l'un de ces protocoles (FTP) afin de mieux comprendre son trafic.



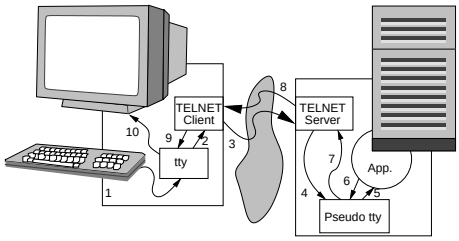
1 Exercices d'échauffement (sans machine)

1. Qu'est-ce qu'un protocole applicatif ?
2. Quels programmes accédant au réseau utilisez-vous couramment ? Savez-vous quels sont les protocoles applicatifs que ces programmes utilisent ?
3. Sur quel modèle de communication s'appuient principalement les applications actuelles ? Comment identifier les rôles des participants pour les applications citées précédemment ?
4. Décrivez les grandes catégories d'applications utilisant les réseaux. Pour chacune d'elles, indiquez les besoins en termes de débit, de tolérance à la variation ce débit, de sensibilité aux pertes et de contraintes temporelles.

2 Connexion à distance

2.1 Rappels

1. Quelle est l'utilité des applications de connexion à distance *(remote login)* ?
2. Quels types d'informations sont échangés par ce genre d'application ?
3. Quelles contraintes peut poser ce type d'application ? Citez des exemples.
4. Quel type de service réseau doivent utiliser ces applications ?



2.2 Protocole TELNET

TELNET facilite la connexion à distance. Il est l'un des protocoles les plus anciens de l'environnement TCP/IP. (Le RFC 854 a été publié en 1983.) Il doit donc fonctionner avec un existant très important de machines et de terminaux et permettre la négociation de nombreux paramètres optionnels (généralement à l'ouverture de la communication) pour s'adapter aux besoins des deux extrémités.

L'hétérogénéité potentielle des deux hôtes impliqués dans l'échange nécessite un service de terminal virtuel, c'est-à-dire un encodage commun à travers le NVT (encodage proche de l'ASCII 7bits pour les caractères imprimables), évitant d'avoir à connaître la correspondance des caractères entre chaque type de destinataires (ce point est important car le NVT est également l'encodage habituel des applications textuelles de TCP/IP).

Le protocole TELNET repose sur une connexion réseau TCP (port serveur **23**) afin de garantir la fiabilité de l'échange. Le contrôle est dit *In-band* : les données circulent dans la même connexion que les informations de négociation. Comme la plupart des protocoles anciens de TCP/IP, il n'intègre aucun mécanisme de sécurité (pas de confidentialité).

Dans l'analyse qui va suivre, vous allez analyser les deux phases caractéristiques du protocole TELNET : négociation puis échange de données.

2.2.1 Capture d'un trafic TELNET

Cette première capture a pour but de percevoir la nature du trafic TELNET. Sur la plateforme d'expérimentation, la topologie 1 a été configurée (postes clients et serveur sur le même LAN). Réalisez la capture de trafic TELNET à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles
  - fenêtre 1 (hôte "client") : tapez `ssh -X etudiant@10.0.7.`**N**1
  - fenêtre 2 (hôte "sonde") : tapez `ssh -X root@10.0.7.`**N**2 (attention, vous êtes administrateur)
  - fenêtre 3 (hôte "serveur") : tapez `ssh -X etudiant@10.0.7.`**N**3
- Vérifiez que TELNET tourne sur le VM serveur (fenêtre 3)
  - recherchez le processus du serveur TELNET, tapez `ps aux | grep telnetd` ou `inetd` (avec la configuration adéquat du service TELNET dans le fichier `/etc/inetd.conf`)
  - visualisez la configuration des interfaces, en particulier celle sur le LAN d'étude (`/sbin/ifconfig eth1`) pour vérifier l'adresse IP du serveur pour la connexion du client (devrait être `10.`**N**`.1.`**N**3)
- Démarrez la capture en lançant l'analyseur sur `10.0.7.`**N**2 (fenêtre 2)
  - lancez l'analyseur, tappez : `wireshark`
  - initiez la capture sur l'interface `eth1`, comme indiqué précédemment
- Démarrez un client TELNET sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en établissant une connexion vers le serveur, tapez : `telnet 10.`**N**`.1.`**N**3
  - identifiez-vous avec le login `etudiant` et le mot de passe correspondant puis tapez quelques commandes UNIX avant de terminer votre session TELNET (fermeture de la session par la commande `exit`)
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à TELNET** (filtre = `telnet`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

2.2.2 Analyse de la négociation TELNET

Les négociations ont principalement lieu au début de la connexion. Elles sont composées de commandes qui peuvent être envoyées dans chaque sens. Chaque commande démarre par la **commande d'échappement** qui est codée sur 1 octet (début d'une commande) : `IAC`=`0xff`. Les **commandes de négociation** d'option (sur 1 octet) sont immédiatement suivies de la valeur de l'option (sur 1 octet) : `WILL`=`0xfb` (indique ce qu'une entité va faire), `WONT`=`0xfc` (ne pas faire), `DO`=`0xfd` (demande à l'autre entité de faire), `DON'T`=`0xfe` (demande de ne pas faire). Exemple :

`IAC`, `DO`, `24`

Les **sous-options** sont transmises (après une demande à l'aide d'un `WILL` puis d'une confirmation avec un `DO`) entre les deux commandes (sur 1 octet) suivantes : `SB`=`0xfa` et `SE`=`0xf0`. Une sous-option se compose du code de l'option sur 1 octet, 1 octet nul puis la valeur de l'option. Exemple :

`IAC`, `SB`, `24`, `0`, `'V'`, `'T'`, `'2'`, `'2'`, `'0'`, `IAC`, `SE`

$(Value)_{10}$	**Option name**
1	Echo
3	Suppress Go Ahead
5	Status
6	Timing Mark
24	Terminal Type
31	Negotiate About Window Size
32	Terminal Speed
33	Remote Flow Control
34	Linemode
35	X Display Location
36	Environment variables
39	New Environment Option
...	

Dans la capture réalisée, essayez de trouver la signification des différents paramètres négociés en observant seulement la partie présentant les octets en hexadécimal de `wireshark`.

1. Analysez les différentes options et sous-options échangées.
2. Trouvez combien de temps dure la négociation.

2.2.3 Analyse de l'échange de données TELNET

Passez les quelques trames de négociation.

1. Quand démarre l'émission de données TELNET ?
2. Quelles transmissions sur le réseau la frappe d'un caractère par l'utilisateur génère-t-elle lors d'une session TELNET ?
3. Que pensez-vous de l'efficacité du protocole ?
4. Quel est le degrès d'interactivité ?
5. Quelles informations sont véhiculées dans l'échange de données ?

2.2.4 Trace TELNET longue distance (facultatif... à traiter de manière autonome si vous êtes nettement en avance par rapport au reste du groupe)

A partir d'une trace contenant une heure de trafic longue distance entre le *Lawrence Berkeley Laboratory* et le reste du monde en janvier 1994, retrouvez des exemples de communications TELNET.

Ces traces , initialement au format `tcpdump` (le même que celui par défaut de `wireshark`), ont été converties en ASCII en prenant soin de renuméroter les adresses IP et de supprimer le contenu des paquet[1].

En voici un extrait :

```
8.430376 22 21 23 33281 1
8.437539 3 4 3930 119 47
8.442644 4 3 119 3930 15
8.454895 26 11 4890 23 1
8.459398 5 2 14037 23 0
8.469004 4 23 4464 119 512
```

La première colonne contient une estampille temporelle relative au début de la capture (exprimée en secondes), les deux colonnes suivantes sont les adresses sources et destinations renumérotées par ordre d'apparition, ensuite se trouvent les numéros de port puis la taille des données (en octets).

Chargez la trace `tme2-lbl.txt.gz`, soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, vers un répertoire **local** (ex : `/tmp`)[2]. Puis à l'aide des outils UNIX standard (`awk`, `perl`, `sed`...), isolez un des flots TELNET et identifiez ses caractéristiques typiques. **Ne demandez pas à votre encadrant d'aide sur ces outils, il est là pour répondre à vos questions liées au réseau**.

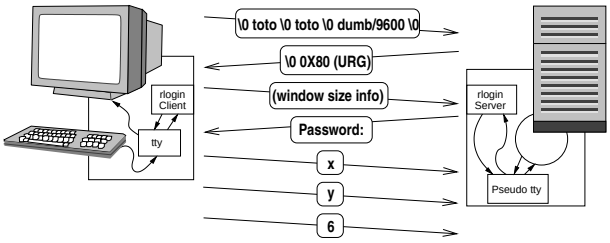
1. Réalisez un chronogramme rapide de quelques échanges TELNET se trouvant dans la trace. Que pouvez-vous dire de l'interactivité ?
2. La sonde est-elle proche de l'émetteur ?
3. Pouvez-vous faire des hypothèses sur le type d'informations échangées ?

2.2.5 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-tel.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

[1] The trace lbl-pkt-4 ran from 14 :00 to 15 :00 on Friday, January 21, 1994 (times are Pacific Standard Time) and captured 1.3 million TCP packets, the dropping about 0.0007 of the total. The tracing was done on the Ethernet DMZ network over which flows all traffic into or out of the Lawrence Berkeley Laboratory, located in Berkeley, California. The raw trace was made using tcpdump on a Sun Sparcstation using the BPF kernel packet filter. Timestamps have microsecond precision. The trace has been "sanitized" using the sanitize scripts. This means that the host IP addresses have been renumbered, and all packet contents removed. The trace was made by Vern Paxson (vern@ee.lbl.gov). The trace may be freely redistributed.

[2] La taille de la trace étant particulièrement importante, si vous travaillez sur votre compte qui est monté par NFS vous obtiendrez des temps de réponse très mauvais.



2.3 Protocole RLOGIN

Le protocole RLOGIN est un autre protocole pour établir des connexions de contrôle à distance. Il est intimement lié au monde UNIX avec les RPC *(Remote Procedure Call)*. Il est également beaucoup plus simple que TELNET : pas de négociation initiale, quelques commandes *in-band* pour, par exemple, redimensionner la taille de la fenêtre. Pour les besoins de fiabilité, une connexion TCP vers le port serveur **513** est utilisée. Aucun mécanisme n'assure la confidentialité de l'échange.

Dans l'analyse qui va suivre, vous allez étudier le protocole RLOGIN en essayant d'effectuer les mêmes interactions qu'avec TELNET au niveau de l'utilisateur.

2.3.1 Capture d'un trafic RLOGIN

Cette seconde capture a pour but de percevoir la nature du trafic RLOGIN. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic RLOGIN à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles (si ce n'est déjà fait).
- Vérifiez que le serveur RLOGIN (`rlogind`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Démarrez un client RLOGIN sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en établissant une connexion vers le serveur, tapez : `rlogin 10.`**N**`.1.`**N**3 (adresse IPv4 sur le réseau d'expérimentation)
  - identifiez-vous avec le login `etudiant` et le mot de passe correspondant puis tapez les quelques commandes exécutées précédemment lors de la capture TELNET
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à RLOGIN** (filtre = `rlogin`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

2.3.2 Analyse de l'échange RLOGIN

1. Quelles informations sont véhiculées à travers le premier échange ?
2. Quelles différences constatez-vous entre RLOGIN et TELNET pour les données applicatives (l'échange réalisé est censé être identique à la précédente capture au niveau des données utilisateur) ?
3. Quelle différence peut-on observer au niveau de l'identification entre RLOGIN et TELNET ?

2.3.3 Trace RLOGIN longue distance (facultatif... à traiter de manière autonome si vous êtes en avance par rapport au reste du groupe)

1. Toujours par rapport à la trace `tme2-lbl.txt.gz` chargée précédemment, identifiez des communications RLOGIN.

2.3.4 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-rlo.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

2.4 Protocole SSH

Ce protocole remplace généralement TELNET et RLOGIN car c'est un protocole de connexion à distance intégrant des mécanismes de sécurité : SSH garantit l'authentification, la confidentialité et l'intégrité des communications. SSH utilise une connexion TCP sur le port serveur **22**.

De nombreuses possibilités d'utilisation sont associées à SSH pour profiter de ses mécanismes de sécurité : multiplexage de plusieurs flux dans une connexion, utilisation d'une connexion SSH comme couche transport pour d'autres applications (vous pouvez, par exemple, créer une connexion SSH entre votre machine résidentielle et le serveur d'accès de l'université, non seulement pour réaliser une connexion à distance textuelle, mais également pour rediriger du trafic entre un client applicatif local et un serveur du centre de calcul de l'université)...

Dans l'analyse qui va suivre, vous allez étudier le protocole SSH en essayant d'effectuer les mêmes interactions que précédemment au niveau de l'utilisateur.

2.4.1 Capture d'un trafic SSH

Cette troisième capture a pour but de percevoir la nature du trafic SSH. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic SSH à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur SSH (`sshd`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Démarrez un client SSH sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en établissant une connexion vers le serveur, tapez : `ssh 10.`**N**`.1.`**N**3 (réseau d'expérimentation)
  - identifiez-vous avec le login `etudiant` et le mot de passe correspondant puis tapez les quelques commandes exécutées précédemment lors des captures TELNET et RLOGIN
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à SSH** (filtre = `ssh`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

2.4.2 Analyse de l'échange SSH

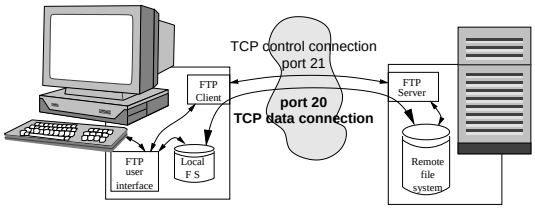
1. Qu'observez-vous au début de l'échange ?
2. Pouvez-vous observer des différences avec TELNET et RLOGIN (l'échange réalisé est identique aux précédents en terme de données utilisateur) ?

2.4.3 Trace SSH longue distance (facultatif... à traiter de manière autonome si vous êtes en avance par rapport au reste du groupe)

1. Toujours par rapport à la trace `tme2-lbl.txt.gz` chargée précédemment, identifiez des communications SSH.

2.4.4 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-ssh.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).



3 Transfert de fichiers

3.1 Protocole FTP

3.1.1 Etude du RFC 959 (sans machine)

Dans cette section nous allons travailler sur un **RFC** *(Request For Comments)* produit par les groupes de travail de l'**IETF** *(Internet Engineering Task Force)* afin d'assurer la standardisation des protocoles de l'Internet. Le **RFC 959** présente une forme classique et donne un aperçu de ce type de document. Récupérez le **RFC 959** sur le site dédié de l'IETF :

- démarrez un navigateur et accédez à la page `http://www.rfc-editor.org/`
- cliquez sur RFC SEARCH et précisez le terme "FTP" pour démarrer la recherche
- sélectionnez le document **RFC 959** dans le résultat de la recherche

Ouvrez ce document et parcourez en rapidement le contenu, puis répondez aux questions suivantes :

1. Que pouvez-vous dire sur la forme du document ? Quelles sont les différentes sections abordées dans ce document ?
2. Précisez l'architecture de communication de FTP. Pourquoi dit-on que les informations de contrôle circulent "hors-bande" ?
3. Quelles sont les différentes commandes à la disposition du client ?
4. Pouvez-vous citer les différents types d'erreur que peut signaler FTP ? Comment les informations d'erreur sont-elles transmises ?

3.1.2 Capture d'un trafic FTP

Cette capture a pour but de percevoir la nature du trafic FTP. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic FTP à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur FTP (`ftpd`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Démarrez un client FTP sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en établissant une connexion vers le serveur, tapez : `ftp 10.`**N**`.1.`**N**3 (réseau d'expérimentation)
  - identifiez-vous avec le login `etudiant` et le mot de passe correspondant
  - déplacez-vous dans le système de fichiers du serveur (commandes de l'interface utilisateur `pwd`, `cd` et `dir`)
  - choisissez un fichier et transférez le sur la machine client (commandes de l'interface utilisateur `get`)
  - terminez l'échange (commandes de l'interface utilisateur `quit`)
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à FTP** (filtre = `ftp or ftp-data`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

3.1.3 Analyse de la connexion FTP de contrôle

Retrouvez la correspondance entre les messages échangés sur le réseau (sur la connexion de contrôle de FTP) et ceux affichés par l'application (interface utilisateur sur le client).

1. Par définition, qui initie la communication entre le client et serveur ? Peut-on l'observer dans la capture ?
2. Quelle commande du protocole FTP identifie l'utilisateur ? Dans la capture, quelle information pouvez-vous observer sur celui-ci ?
3. Quelle est la commande qui authentifie ensuite l'utilisateur ? Le mot de passe apparaît-il en clair sur le réseau ?
4. Quel est l'intérêt de la commande suivant l'authentification ?
5. A quoi sert la commande `PORT` ? Analysez ses paramètres. Pourquoi est-elle émise à ce point de l'échange ?
6. La commande `LIST` permet d'obtenir la liste des fichiers du répertoire courant au niveau du serveur. Pourquoi est-elle suivie de deux messages envoyés par le serveur ?
7. Quelles sont les autres commandes que vous observez ? A quoi servent-elles ?
8. A quels moments de la transaction ont lieu les transferts de fichiers ?

3.1.4 Analyse de la connexion FTP de données

1. A quoi correspondent les données échangées sur les connexions de tranfert de données ?
2. Sur quels ports ces données sont-elles envoyées ?
3. Quelle synchronisation observez-vous entre les messages sur la connexion de contrôle et ceux de la connexion de données ?

3.1.5 Trace FTP longue distance (facultatif... à traiter de manière autonome si vous êtes en avance par rapport au reste du groupe)

1. Toujours par rapport à la trace `tme2-lbl.txt.gz` chargée précédemment, identifiez des communications FTP avec les connexions `FTP` et `FTP-DATA` associées.
2. Tracez le chronogramme.
3. Que pouvez vous dire de l'interactivité par rapport à TELNET ?

3.1.6 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-ftp.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

3.2 Protocoles SCP/SFTP

Plusieurs protocoles permettent la transmission de fichier sécurisée. L'application `scp` est un client de la suite logicielle associée à SSH (transfert de fichiers à un serveur `sshd`). `scp` s'utilise de manière similaire à `rcp` (copie à distance de la famille des `r*`-commandes UNIX). Il existe également le client `sftp` qui conserve un mode de fonctionnement similaire à FTP dans un tunnel SSH (un serveur spécifique `sftp-server` lui est dédié).

3.2.1 Capture d'un trafic SCP/SFTP

Cette capture a pour but de percevoir la nature du trafic associé à un transfert de fichier avec un protocole sécurisé. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic SCP ou SFTP[3] à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur SCP (`sshd`) ou SFTP (`sftp-server`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Démarrez un client SCP ou SFTP sur `10.0.7.`**N**1 et récupérez à partir du serveur le fichier précédemment transféré (fenêtre 1)

  SCP tapez : `scp etudaint@10.`**N**`.1.`**N**`3:<fichier_distant> <fichier_local>` puis authentifiez-vous

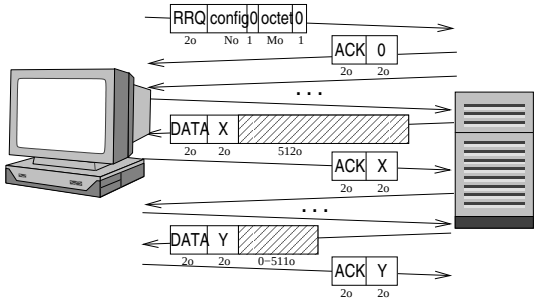
  SFTP tapez : `sftp 10.`**N**`.1.`**N**3 puis authentifiez-vous et tapez les quelques commandes exécutées précédemment lors de la capture FTP
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à SCP ou SFTP** (filtre = `ssh`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

3.2.2 Analyse de l'échange SCP/SFTP

1. Rappelez le fonctionnement des protocoles SCP et SFTP.
2. Pouvez-vous faire la correspondance entre les messages du terminal et les trames échangées ?
3. Quelles différences constatez-vous avec FTP ?

3.2.3 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-scp.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).



3.3 Protocole TFTP

3.3.1 Capture d'un trafic TFTP

Cette dernière capture a pour but de percevoir la nature du trafic TFTP. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic TFTP à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur TFTP (`tftpd`) tourne sur `10.0.7.`**N**3 et qu'un répertoire est configuré pour réaliser les transferts (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)

[3] Nous notons SCP ou SFTP les trafics associé aux applications `scp` ou `sftp`.

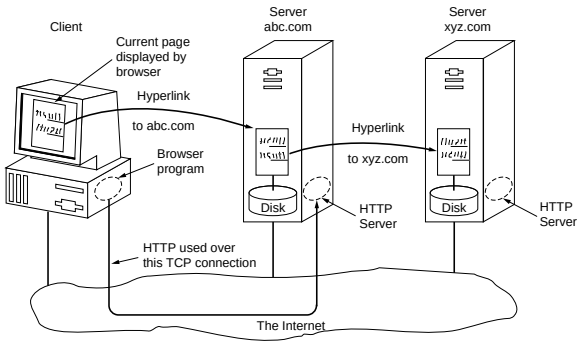
- Démarrez un client TFTP sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en accédant au serveur, tapez : `tftp 10.`**N**`.1.`**N**3 (réseau d'expérimentation)
  - envoyez un fichier au serveur avec la commande `put`
  - terminez l'échange avec la commande `quit`
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à TFTP** (filtre = `tftp`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

3.3.2 Analyse de l'échange TFTP

1. Rappelez le fonctionnement du protocole TFTP.
2. Pouvez-vous faire la correspondance entre les messages du terminal et les trames échangées ?
3. Quelles différences constatez-vous avec FTP ?

3.3.3 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-tft.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).



4 Trafic web

4.1 Exercices (sans machine)

1. Expliquez les étapes nécessaires à la récupération d'une page web. Supposez que vous souhaitez récupérer une page composée d'un fichier HTML indiquant deux objets de taille réduite stockés sur le même serveur. En négligeant les temps de transmission de ces objets, indiquez le délai nécessaire pour obtenir la page. Illustrez vos réponses avec un chronogramme.
2. Quelles optimisations prévues par HTTP 1.1 utilisent les serveurs web actuels pour réduire la latence des échanges ? En reprenant l'exemple précédent, illustrez vos réponses avec des chronogrammes.
3. Une autre possibilité pour réduire le temps de réponse est l'utilisation de la mise en mémoire cache. Décrivez où intervient ce mécanisme et pour quels types d'objets il est intéressant.

4.2 Protocole HTTP

4.2.1 Capture d'un trafic HTTP

Cette dernière capture a pour but de percevoir la nature du trafic HTTP. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic SSH à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur HTTP (`apache2`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)

- Démarrez un client HTTP sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client de votre choix (`firefox`...)
  - établissant une connexion vers le serveur en spécifiant l'URL suivante : `http://10.`**N**`.1.`**N**3 (accès via le réseau d'expérimentation)
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à HTTP** (filtre = `http`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

4.2.2 Analyse de l'échange HTTP

1. Observez-vous le mécanisme de récupération d'une page présenté précédemment ?
2. Quels paramètres sont négociés entre le client et le serveur ?
3. Quelles optimisations sont mises en œuvre pour accélérer le rapatriement des pages web ?
4. Pouvez-vous afficher la page web à partir de la capture ?

4.2.3 Création d'une nouvelle page web (facultatif... à traiter si vous êtes en avance)

Pour capturer un peu plus d'informations relatives aux échanges HTTP, vous pouvez créer des pages web sur la VM "serveur".

Dans le répertoire `/var/www`, démarrez en créant une page web intégrant du texte et 3 images. Vous devez donc avoir les fichiers suivants :

- 1 fichier HTML avec un nom différent de `index.html` qui contiendra le texte et référencera les images (voir sur le web comment taper les quelques lignes de HTML nécessaires)
- 3 images référencées dans le fichier HTML ci-dessus (à tranférer via SCP du poste ARI vers le répertoire `/var/www` en tant qu'utilisateur `root`).

Vous pouvez accéder à votre nouvelle page (par exemple : `NewPage.html`) à partir du client en spécifiant l'URL suivante :

`http://10.`**N**`.1.`**N**`3/NewPage.html`

Réalisez à nouveau la capture de la partie 4.2.1 en accédant cette fois à la page que vous venez de créer et répondez ensuite aux questions de la partie 4.2.2.

4.2.4 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme2-htt.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

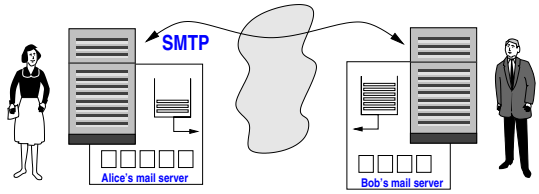
5 Avant de quitter la salle

- Si vous avez enregistré des captures sur la VM "sonde", n'oubliez pas de les rapatrier sur votre compte utilisateur de l'ARI. Tapez : `scp root@10.0.7.`**N**`2:<ma_trace> <destination_locale>`.
- Avant de terminer vos connexions sur les équipements de la plateforme, supprimez vos fichiers et modifications effectuées afin de retrouver l'état initial.

ARES - Lab n°3

Couche application (2) : Messagerie, DNS et SNMP

Ce support est le second consacré à la couche application. Il propose quelques exercices à réaliser sans machine et l'étude des protocoles des applications suivantes : SMTP, POP, IMAP, DNS et SNMP. Pour illustrer nos propos, le trafic réseau de chacune de ces applications sera capturé et analysé avec l'outil `wireshark`.



1 Messagerie

Nous allons détailler les différents mécanismes et protocoles associés à l'émission et à la réception d'un message électronique. En particulier, nous étudierons deux approches qui diffèrent au niveau de la proximité du client : soit basée sur SMTP/POP/IMAP (client local), soit sur HTTP *(web-mail)*.

1.1 Quelques exercices (sans-machine)

1. Bob accède à sa messagerie par le web. Il envoie un message à Alice. Cette dernière rapatrie ses messages sur son ordinateur de bureau lorsque celui-ci est allumé. Décrivez les échanges d'informations ainsi que les protocoles mis en jeu.

2. Rappelez la structure du message échangé par les serveurs de courrier électronique.

3. Pouvez-vous décoder ce champ d'en-tête ?
   `Subject: =?iso-8859-1?B?Qydlc3QgcGFzIGZhY2lsZSAhCg==?=`

4. Vous envoyez un courrier électronique avec un message en texte et en HTML, accompagné de quelques pièces jointes : une image au format PNG, du son encodé en MP3 et un fichier WAD pour Doom. Quelles lignes d'en-tête devrait-on observer dans le message ?

1.2 Services de messagerie sur la plateforme d'experimentation

La plateforme permet d'accéder aux services de messagerie via les protocoles SMTP, POP, IMAP et HTTP *(web-mail)*.

Nous supposons que vous êtes sur le poste ARI **N**. L'envoi d' *e-mail* se fait par le biais du protocole SMTP qui contacte le MTA localisé sur la VM "serveur". Ce dernier peut être identifié grâce à son nom (`mail.etu`**N**`.plateforme.lan`) ou son adresse IPv4 (`10.`**N**`.1.`**N**`3`). Vous pouvez utiliser la boîte *e-mail* `etudiant` sur le MTA afin de pouvoir recevoir des messages[1].

Vous pouvez alors envoyer des messages à partir d'un premier compte dédié local au client (par exemple "`Test SMTP`") simplement en configurant votre agent de messagerie (`evolution` ou autre UA disponible). Cette configuration doit comprendre comme serveur de messagerie `mail.etu`**N**`.plateforme.lan` et SMTP (ou ESMTP) comme protocole d'envoi des *e-mails*.

Concernant la récupération des messages, vous pouvez accéder à votre boîte `etudiant@etu`**N**`.plateforme.lan` avec les deux protocoles suivant : POP ou IMAP. Pour cela, un deuxième compte local "`Test POP`" est a configurer pour un accès et un retrait des messages par le protocole POP. Un troisième compte local, "`Test IMAP`", est également à configurer pour un accès via le protocole IMAP aux messages présents sur le serveur.

1.2.1 Configuration des serveurs

Voici les 3 serveurs utilisés pour le lab :

- `smtpd` : C'est le serveur **Postfix**[2], MTA alternatif à Sendmail. Pour éviter de changer la configuration du serveur, il est nécessaire d'utiliser la boîte associée au compte UNIX `etudiant`.

[1] En configuration de base, un serveur UNIX associe ses comptes utilisateur UNIX a des boîtes *e-mail*.
[2] `http://www.postfix.org/`

- `imapd` : C'est le serveur **Courier-IMAP**[3] qui assure le service POP et IMAP.
- `apache2` : C'est le serveur HTTP, sur lequel repose **Squirrelmail**[4] (écrit complètement en PHP, sortie en pur HTML).

1.2.2 Configuration des clients

Sur votre machine client, l'utilisation du UA `evolution` nécessite quelques configurations pour rajouter des comptes locaux ("`Test SMTP`", "`Test POP`" et "`Test IMAP`"). Après avoir lancé l'application, dans le menu Edition, sélectionnez Preférence puis utilisez le bouton + Ajouter pour ajouter un nouveau compte. **Attention, les comptes sont peut-être déjà configurés,** dans ce cas effacez les *e-mails* présents et vérifiez les paramètres utilisés.

"Test SMTP" Les étapes pour ajouter dans `evolution` le compte local "`Test SMTP`" pour émettre des messages en SMTP sont les suivantes :

- Ne restaurez pas la session si ce choix vous est proposé
- Configurez le nom "`etudiant sur la VM3`" et l'adresse *e-mail* "`etudiant@mail.etu`**N**`.plateforme.lan`" (vous pouvez préciser que c'est le compte par défaut)
- Ne configurez pas de serveur de réception (choisir "Aucun")
- Configurez le serveur SMTP (sélectionnez le type SMTP, précisez le serveur "`mail.etu`**N**`.plateforme.lan`" et ne sélectionnez aucune authentification)
- Terminez en configurant le nom "`Test SMTP`"

"Test POP" Les étapes pour ajouter dans `evolution` le compte local "`Test POP`" pour récupérer via POP les messages de la boîte `etudiant` du serveur sont les suivantes :

- Utilisez à nouveau le nom "`etudiant sur la VM3`" et l'adresse *e-mail* "`etudiant@mail.etu`**N**`.plateforme.lan`"
- Configurez le serveur POP (sélectionnez le type POP, préciser le serveur "`mail.etu`**N**`.plateforme.lan`", le nom d'utilisateur "`etudiant`" et ne modifiez aucun mécanismes de sécurité ou d'authentification)
- Indiquez que vous souhaitez conserver les messages sur le serveur et laissez les valeurs par défaut des autres options de réception
- Ne configurez pas de serveur SMTP (sélectionnez "`Sendmail`")
- Terminez en configurant le nom "`Test POP`"

"Test IMAP" Les étapes pour ajouter dans `evolution` le compte local "`Test IMAP`" pour accéder via IMAP aux messages de la boîte `etudiant` du serveur sont les suivantes :

- Utilisez à nouveau le nom "`etudiant sur la VM3`" et l'adresse *e-mail* "`etudiant@mail.etu`**N**`.plateforme.lan`"
- Configurez le serveur IMAP (sélectionnez le type IMAP, préciser le serveur "`mail.etu`**N**`.plateforme.lan`", le nom d'utilisateur "`etudiant`" et ne modifiez aucun mécanismes de sécurité ou d'authentification)
- Laisser les valeurs par défaut des autres options de réception
- Ne configurez pas de serveur SMTP (sélectionnez "`Sendmail`")
- Terminez en configurant le nom "`Test IMAP`"

Pour le service de messagerie par *web-mail*, vous devez accéder aux comptes via le navigateur web en vous connectant au serveur à travers **SquirrelMail** via l'URL `http://mail.etu`**N**`.plateforme.lan/squirrelmail`. Entrez alors le nom du compte (`etudiant`) sans spécifier le domaine du serveur mail.

[3] `http://www.courier-mta.org/imap/`
[4] `http://squirrelmail.org/`

1.2.3 Préparation pour réaliser les captures

Les captures que vous allez réaliser dans la prochaine section ont pour but de percevoir la nature des trafics de messagerie. Sur la plateforme d'expérimentation, la topologie 1 a été configurée (postes clients et serveur sur le même LAN). Vous réaliserez ces captures de trafic à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles
  - fenêtre 1 (hôte "client") : tapez `ssh -X etudiant@10.0.7.`**N**1
  - fenêtre 2 (hôte "sonde") : tapez `ssh -X root@10.0.7.`**N**2 (attention, vous êtes administrateur)
  - fenêtre 3 (hôte "serveur") : tapez `ssh -X etudiant@10.0.7.`**N**3
- Vérifiez que les serveurs Postfix, Courier-IMAP et HTTP tournent sur `10.0.7.`**N**3 (fenêtre 3)
  - rechercher les processus des serveurs, tapez `ps aux | grep postfix` ou `imapd` ou `apache2`
  - visualisez la configuration des interfaces, en particulier celle sur le LAN d'étude (`/sbin/ifconfig eth1`) pour vérifier l'adresse IPv4 du serveur pour la connexion du client (devrait être `10.`**N**`.1.`**N**3)
- Démarrez la capture en lançant l'analyseur sur `10.0.7.`**N**2 (fenêtre 2)
  - lancez l'analyseur, tappez : `wireshark`
  - initier la capture sur l'interface `eth1`, comme indiqué précédemment
- Démarrez un UA, `telnet` ou un navigateur sur `10.0.7.`**N**1 (fenêtre 1)
  - soit tapez `evolution` et utilisez-le en fonction des captures demandées relatives à SMTP/POP/IMAP
  - soit tapez `telnet mail.etu`**N**`.plateforme.lan 25`
  - soit tapez `firefox` et entrez l'URL `http://mail.etu`**N**`.plateforme.lan/squirrelmail`
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez les captures. **Filtrez les trafics afin de ne conserver que ceux relatifs à SMTP, POP, IMAP ou HTTP** (filtre = `smtp`, `pop`, `imap` ou `http`). Enregistrez les traces filtrées pour pouvoir les ré-utiliser ultérieurement.

1.3 Emission du message

1.3.1 Envoi avec le protocole SMTP

Réalisez une capture en utilisant l'UA `evolution` pour envoyer un *e-mail* à partir du compte local par défaut (`"Test SMTP"`) vers la boîte (`etudiant`) du serveur.

1. Quelles sont les commandes du protocole SMTP observées lors de l'émission d'un courrier ? Pouvez-vous indiquer leur utilité et le type de réponse produite ?
2. Quelles sont les contraintes imposées à la forme du courrier ? Expliquez la structure de ce dernier et détaillez les champs qui composent son en-tête.
3. Que pensez-vous des possibilités d'identification du protocole SMTP ?

Sans la plateforme... En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme3-smt.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

1.3.2 Envoi avec le protocole TELNET

Une alternative, moins attractive mais très efficace, est l'accès au serveur SMTP par le client `telnet`. Il suffit de taper la commande : `telnet mail.etu`**N**`.plateforme.lan 25`.

1. Vérifiez que vous pouvez envoyer un *e-mail* de cette manière.
2. Si vous réalisez une capture, quels filtres allez-vous utiliser ?

1.3.3 Envoi avec le protocole HTTP

Réalisez une capture en utilisant le *web-mail* pour envoyer un *e-mail* vers votre boîte (`etudiant`) sur le serveur.

1. Pouvez-vous retrouver le message original dans la réponse du serveur ?
2. Que pensez-vous de la confidentialité lorsque vous consultez votre courrier de cette manière ?

Sans la plateforme... En cas de besoin, vous pouvez télécharger et analyser la trace `tme3-wm1.dmp` (similaire à celle capturée précédemment) des localisations habituelles (voir la partie 1.3.1).

1.4 Réception du message

1.4.1 Réception avec le protocole POP

Réalisez une capture en utilisant le UA `evolution` pour recevoir un *e-mail* sur le compte local "`Test POP`" (assurez-vous qu'un *e-mail* a bien été reçu sur ce compte précédemment).

1. Quelles sont les commandes utilisées par le protocole POP lors de la récupération d'un courrier ? Pouvez-vous indiquer leur utilité et le type de réponse produit ?
2. A votre avis, quelles seraient les réponses du serveur POP s'il y avait plusieurs messages en attente ?
3. Quelles sont les différences entre le message envoyé précédemment et celui reçu ici ?

Sans la plateforme... En cas de besoin, vous pouvez télécharger et analyser la trace `tme3-pop.dmp` (similaire à celle capturée précédemment) des localisations habituelles (voir la partie 1.3.1).

1.4.2 Réception acec le protocole IMAP

Réalisez une capture en utilisant le UA `evolution` pour recevoir un *e-mail* sur le compte local "`Test IMAP`" (assurez-vous qu'un *e-mail* a bien été reçu sur ce compte précédemment).

1. Quels types d'échanges sont réalisés entre le client et le serveur IMAP ?
2. Quelles différences protocolaires observez-vous entre POP et IMAP ?
3. Quelles sont les différences entre le message envoyé précédemment et celui reçu ici ?
4. Pensez vous que l'authentification soit plus sécurisée avec IMAP ?

Sans la plateforme... En cas de besoin, vous pouvez télécharger et analyser la trace `tme3-ima.dmp` (similaire à celle capturée précédemment) des localisations habituelles (voir la partie 1.3.1).

1.4.3 Réception acec le protocole TELNET

POP et IMAP étant des protocoles textuels, le client `telnet` est utilisable pour se connecter vers les serveurs POP ou IMAP. Il suffit de taper la commande : `telnet mail.etu`**N**`.plateforme.lan <portnum>`. Le `<portnum>` correspond au numéro de port serveur du protocole utilisé : 110 pour POP et 143 pour IMAP.

1. Vérifiez les actions que vous pouvez réaliser de cette manière avec POP.
2. Vérifiez les actions que vous pouvez réaliser de cette manière avec IMAP.
3. Si vous réalisez une capture, quels filtres allez-vous utiliser ?

1.4.4 Réception acec le protocole HTTP

Réalisez une capture en utilisant le *web-mail* pour consulter un *e-mail* sur le compte du serveur (assurez-vous qu'un *e-mail* a bien été reçu sur ce compte précédemment).

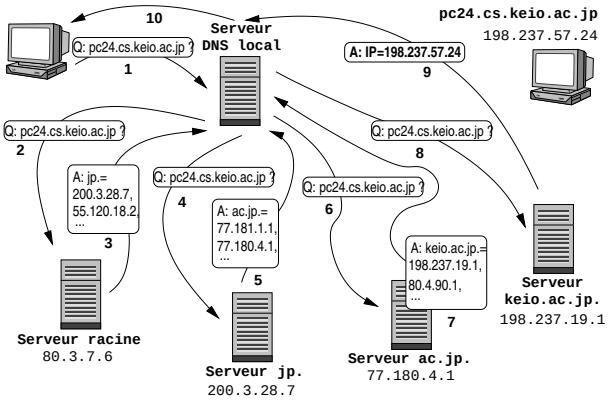
1. Pouvez-vous retrouver le message original dans la réponse du serveur ?
2. La consultation du message retourne beaucoup de trafic HTTP. Discutez des performances d'une consultation à travers le web.

Sans la plateforme... En cas de besoin, vous pouvez télécharger et analyser la trace `tme3-wm2.dmp` (similaire à celle capturée précédemment) des localisations habituelles (voir la partie 1.3.1).

1.5 Trace longue distance (facultatif... à traiter de manière autonome si vous êtes nettement en avance par rapport au reste du groupe)

A partir d'une trace contenant une heure de trafic longue distance entre le *Lawrence Berkeley Laboratory* et le reste du monde en janvier 1994, retrouvez des exemples de communications SMTP, POP et IMAP.

Chargez la trace `tme2-lbl.txt.gz`, soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, vers un répertoire **local** (ex : `/tmp`)[5]. Puis à l'aide des outils UNIX standard (`awk`, `perl`, `sed`...), isolez un des flots intéressants (SMTP, POP et IMAP) et identifiez leurs caractéristiques typiques. **Ne demandez pas à votre encadrant d'aide sur ces outils, il est là pour répondre à vos questions liées au réseau**.



2 Service d'annuaire

2.1 Le système DNS (sans machine)

1. Chaque machine de l'Internet est généralement associée à un serveur de nom local et un serveur de nom de référence (*authoritative*). Quel est le rôle joué par chacun d'eux au sein du système DNS ?

2. A partir d'une machine utilisateur qui comporte un navigateur (client web) et agent de messagerie (client mail), vous souhaitez surfer sur le site web d'une institution (par exemple votre université), puis envoyer un *e-mail* vers le serveur mail de celle-ci. Quelles entités vont avoir à recourir au système DNS, et en particulier, à demander des résolutions qui impliquent le serveur de référence de l'institution ? Le serveur web et le serveur de courrier de cette institution peuvent-ils partager le même nom (par exemple `server.upmc.org`) ?

3. En surfant sur le Web, vous cliquez sur un lien menant à une page qui vous intéresse. Votre machine ne connaît pas l'adresse IP correspondant à l'URL de la page demandée et celle-ci ne se trouve pas dans le cache de votre navigateur. Si n serveurs DNS sont visités de manière **itérative** avant d'obtenir l'adresse IP recherchée, en combien de temps peut-on escompter voir apparaître la page (le temps de transmission de l'objet est négligeable) ? Faites un chronogramme pour illustrer vos réponses.

2.2 Etude manuelle d'un échange DNS (sans machine)

Le système DNS repose sur un échange de messages en mode non connecté. Voici un exemple composé de deux trames que nous vous proposons d'étudier à la main (sans Wireshark). Utilisez directement le support du lab pour entourer les différents champs sur les traces. Attention au codage des noms avec renvoi (code `0xC0`+n indiquant sur un octet la distance n en octet du début du message DNS).

2.2.1 Requête DNS

Voici une trame observée sur le réseau :

```
0000  00 07 e9 0c 90 62 00 20  ed 87 fd e6 08 00 45 00   .....b.  ......E.

0010  00 39 00 00 40 00 40 11  a9 71 84 e3 3d 7a 84 e3   .9..@.@. .q..=z..
```

[5] La taille de la trace étant particulièrement importante, si vous travaillez sur votre compte qui est monté par NFS vous obtiendrez des temps de réponse très mauvais.

```
0020  4a 02 85 05 00 35 00 25  c0 74 a0 71 01 00 00 01   J....5.% .t.q....
0030  00 00 00 00 00 00 03 77  77 77 04 6c 69 70 36 02   .......w ww.lip6.
0040  66 72 00 00 01 00 01                              fr.....
```

1. Analysez manuellement la trame ci-dessus à l'aide du support de cours.
2. Quel est le but du message contenu dans cette trame ? Quelle action de l'utilisateur a pu déclencher cette requête ?

2.2.2 Réponse DNS

Peu de temps après, on peut observer la trame suivante sur le réseau :

```
0000  00 20 ed 87 fd e6 00 07  e9 0c 90 62 08 00 45 00   . ...... ...b..E.
0010  00 cf 2a 2d 00 00 3f 11  bf ae 84 e3 4a 02 84 e3   ..*-..?. ....J...
0020  3d 7a 00 35 85 05 00 bb  a1 3b a0 71 85 80 00 01   =z.5.... .;.q....
0030  00 02 00 03 00 03 03 77  77 77 04 6c 69 70 36 02   .......w ww.lip6.
0040  66 72 00 00 01 00 01 c0  0c 00 05 00 01 00 00 54   fr...... .......T
0050  60 00 08 05 68 6f 72 75  73 c0 10 c0 29 00 01 00   `...horu s...)...
0060  01 00 00 54 60 00 04 84  e3 3c 0d c0 10 00 02 00   ...T`... .<......
0070  01 00 00 54 60 00 07 04  69 73 69 73 c0 10 c0 10   ...T`... isis....
0080  00 02 00 01 00 00 54 60  00 09 06 6f 73 69 72 69   ......T` ...osiri
0090  73 c0 10 c0 10 00 02 00  01 00 00 54 60 00 0e 06   s....... ...T`...
00a0  73 6f 6c 65 69 6c 04 75  76 73 71 c0 15 c0 4d 00   soleil.u vsq...M.
00b0  01 00 01 00 00 54 60 00  04 84 e3 3c 02 c0 60 00   .....T`. ...<..`.
00c0  01 00 01 00 00 54 60 00  04 84 e3 3c 1e c0 75 00   .....T`. ...<..u.
00d0  01 00 01 00 01 16 cb 00  04 c1 33 18 01           ........ ..3..
```

1. Analysez manuellement la trame ci-dessus.
2. Quelles informations sont renvoyées par le serveur DNS local ? Correspondent-elles à celles attendues par le client ?

2.2.3 Vérification des analyses manuelles

Seulement après avoir effectué les deux analyses manuelles ci-dessus, vérifiez les résultats à l'aide du logiciel `wireshark` sur votre poste ARI. Chargez la trace `tme3-dn1.dmp` soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`.

2.3 Le système DNS de la plateforme

Pour les besoins de la plateforme, nous avons installé un serveur DNS sur chaque VM "serveur". Celui-ci joue le rôle de serveur local, de serveur de référence et de relais. Vous trouverez ainsi sur celui-ci les informations relatives à la zone `etu`**N**`.plateforme.lan` (si vous êtes sur du poste ARI **N**) et la résolution inverse.

2.3.1 Configuration du client et du serveur DNS

1. Comment un hôte peut-il accéder au système DNS ? Faut-il utiliser un programme client ? Quels paramètres faut-il configurer ? Etudiez le fichier `/etc/resolv.conf` sur la machine client (`10.0.7.`**N**`1`) et explicitez-en les paramètres.
2. La configuration du serveur **BIND**[6] sur la VM "serveur" (`10.0.7.`**N**`3`) est visible dans le fichier `/etc/bind/named.conf.local`. Celui-ci indique les deux zones contrôlées localement :
   - **etuN.plateforme.lan** décrite dans le fichier `/etc/bind/db.etu`**N**`.plateforme.lan`
   - **N.10.in-addr.arpa** pour la résolution inverse, aussi dans le fichier `/etc/bind/db.etu`**N**`.plateforme.lan`

   Analysez le contenu de ces deux fichiers et expliquez leur utilité. Précisez ce qu'il est nécessaire de modifier si l'on souhaite déclarer une nouvelle machine sur le serveur.
3. Comment générer des échanges DNS ? Citez au moins 4 possibilités que vous testerez dans la capture suivante.

2.3.2 Capture d'échanges DNS locaux

Cette troisième capture a pour but de percevoir la nature du trafic DNS. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic DNS à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur BIND (`named`) tourne sur `10.0.7.`**N**`3` (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**`2` (fenêtre 2)
- Sur la VM "client", vérifier la configuration locale du DNS et réalisez les différentes actions permettant de déclencher des requêtes DNS proposées précédemment (fenêtre 1)
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif au DNS** (filtre = `dns`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

2.3.3 Analyse des échanges DNS locaux

1. Analysez les trames échangées.
2. La résolution locale modifie-t-elle les échanges DNS ?

2.4 Accès au DNS de l'Internet

Le serveur DNS de chaque PC serveur joue le rôle de serveur local pour les zones autres que **etuN.plateforme.lan**. Ce serveur n'ayant pas accès au reste de l'Internet, les requêtes vers les autres zones de l'Internet doivent être relayées vers le serveur DNS de la baie (qui a un accès vers le système DNS global).

[6] BIND *(Berkeley Internet Name Daemon)* : http ://www.isc.org/software/bind

2.4.1 Capture d'échanges DNS externe

Cette troisième capture a pour but de percevoir la nature d'un autre trafic DNS. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic DNS à l'aide du logiciel `wireshark` :

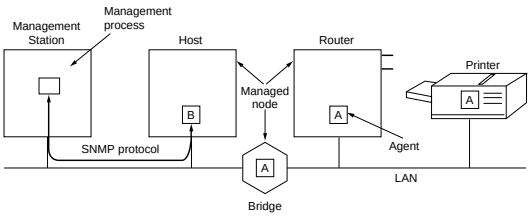
- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur BIND (`named`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Sur la VM client, vérifier la configuration locale du DNS, puis tapez `dig www.apple.com` (fenêtre 1)
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif au DNS** (filtre = `dns`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

2.4.2 Analyse de l'échange DNS externe

1. Analysez rapidement les deux trames contenues dans la trace.
2. Expliquez le but de cet échange.
3. A votre avis, pourquoi la résolution de nom `www.apple.com` est-elle renvoyée vers des serveurs du domaine `aka*.net` ?

2.4.3 Sans la plateforme...

En cas de besoin, vous pouvez télécharger et analyser la trace `tme3-dn2.dmp` (similaire à celle capturée précédemment) des localisations habituelles (voir la partie 1.3.1).



3 Administration réseau

3.1 Exercices sur l'administration réseau (sans machine)

1. Pour un administrateur réseau, quel est l'intérêt d'utiliser des outils de gestion du réseau ? Citez plusieurs possibilités.
2. Représentez sur un schéma intégrant quelques éléments à administrer les mécanismes de bases de l'administration de réseau (éléments applicatifs, messages échangés...).
3. Définissez les termes suivants : Station d'administration, Equipement administré, Agent d'administration, Base d'information de gestion (MIB), Structure des informations de gestion (SMI) et Protocole de gestion du réseau.
4. Quels sont les PDU utilisés par SNMP ? Quels messages sont utilisés pour des requêtes/réponses ou des envois spontanés ? Quel est la différence entre ces deux types d'échanges ? Quels en sont les avantages et les inconvénients ?
5. A votre avis, pour quelles raisons utilise-t-on UDP plutôt que TCP pour le transport des PDU SNMP ?
6. Dans la suite, un administrateur souhaite gérer les routeurs du réseau de son entreprise grâce au protocole SNMP. Ce réseau fonctionne sous TCP/IP et interconnecte plusieurs réseaux locaux à l'aide de routeurs dont le service SNMP est activé.
   (a) Proposez un mécanisme pour découvrir les différentes machines présentes sur le réseau local de la station d'administration.
   (b) Expliquez comment vérifier qu'une machine est bien un routeur (la MIB-II standard définit un objet simple `ipForwarding`).
   (c) Comment obtenir le nom de ces routeurs (la MIB-II standard définit l'objet `system.sysName` de type chaîne de caractère...) ?

(d) Sachant que la MIB-II propose un objet tableau `ipAddrTable` qui référence toutes les interfaces d'une machine avec leurs paramètres IP (addresse IP, masque de réseau, adresse de diffusion...), précisez comment obtenir toutes les adresses IP (champ `ipAdEntAddr`) d'un routeur.

(e) Précisez comment modifier la valeur du masque de réseau (champ `ipAdEntNetMask`) associé à l'interface 3 d'un routeur (les entrées de l'objet table `ipAddrTable` sont indexées par le numéro de cette interface).

(f) Connaissant les informations précédentes disponibles dans la MIB-II, proposez un mécanisme général pour découvrir tous les routeurs du réseau de l'entreprise. Indiquez les limitations de votre approche.

3.2 Protocole SNMP

3.2.1 Capture de trafic SNMP

Cette dernière capture a pour but de percevoir la nature du trafic SNMP. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic SNMP à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur SNMP (`snmpd`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Utilisez les commandes Net-SNMP (`snmpget`, `snmpgetnext`, `snmpwalk`...) sur `10.0.7.`**N**1 (fenêtre 1)
  - tapez `snmpget -v 1 -c public 10.`**N**`.1.`**N**`3 .1.3.6.1.2.1.1.1.0`
  - puis tapez `snmpgetnext -v 1 -c public 10.`**N**`.1.`**N**`3 .1.3.6.1.2.1.1.9.1.3.1`
  - puis tapez `snmpwalk -v 1 -c public 10.`**N**`.1.`**N**`3 .1.3.6.1.2.1.1.9.1.3`
  - puis tapez `snmpset -v 1 -c private 10.`**N**`.1.`**N**`3 .1.3.6.1.2.1.1.4.0 s toto@upmc.fr`
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à SNMP** (filtre = `snmp`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

3.2.2 Analyse de la première requête SNMP

1. Analysez la première trame en détaillant le mécanisme d'encodage de la couche application.
2. Quel est le but de cette requête ?
3. Qui a généré ce message ?

3.2.3 Analyse de la réponse SNMP

1. Suite à l'émission de la trame précédente, une seconde trame est émise. Analysez cette dernière.
2. Quel est le type d'équipement qui a été impliqué ?
3. A la vue de cet échange, que pensez-vous de la sécurité associée à SNMP ?

3.2.4 Analyse du deuxième échange SNMP

1. Après ce premier échange, analysez les deux trames échangées ensuite.
2. Quelle nouvelle opération est réalisée dans cet échange ? Quelles possibilités offre ce type de requête ?

3.2.5 Analyse des échanges SNMP suivants

1. Après ces deux premiers échanges, analysez les trames échangées ensuite.
2. Quelle mécanisme génère ces échanges ?

3.2.6 Analyse du dernier échange SNMP

1. Analysez ensuite la dernière trame émise par le client.
2. Quelle nouvelle opération est réalisée dans cet émission ? Quelles possibilités offre ce type de message ?

3.2.7 Sans la plateforme...

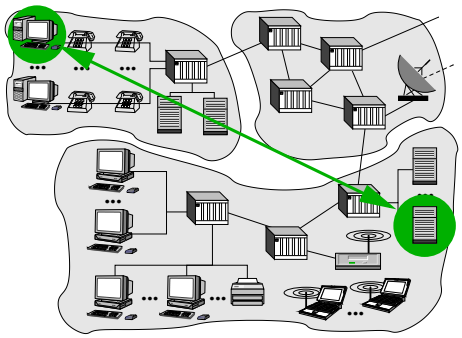
En cas de besoin, vous pouvez télécharger la trace `tme3-snm.dmp` (des localisations habituelles, voir la partie 1.3.1) pour répondre aux questions de la section 3.2.

4 Avant de quitter la salle

- Si vous avez enregistré des captures sur la VM "sonde", n'oubliez pas de les rapatrier sur votre compte utilisateur de l'ARI. Tapez : `scp root@10.0.7.N2:<ma_trace> <destination_locale>`.
- Avant de terminer vos connexions sur les équipements de la plateforme, supprimez vos fichiers et modifications effectuées afin de retrouver l'état initial.

ARES - Lab n°4

Couche transport (1) : TCP et UDP



1 Rappels sur la couche transport (sans machine)

1. Un client web souhaite accéder à un document dont il connaît l'URL. L'adresse IP du serveur concerné est initialement inconnue. Quels protocoles de la couche application sont requis pour satisfaire cette requête ?
2. Quels protocoles de la couche transport sont nécessaires pour satisfaire cette même requête ?

1.1 Protocole en mode non connecté, UDP

1. Rappelez les caractéristiques d'un protocole en mode non connecté.
2. Indiquez les principales caractéristiques du protocole UDP.
3. Pour quelles raisons un développeur pourrait-il préférer le protocole UDP à un autre protocole de transport ?
4. A votre avis, une application peut-elle réaliser un transfert de données fiable si elle repose sur UDP ? Justifiez votre réponse.

1.2 Protocole en mode orienté connexion, TCP

1. Rappelez les caractéristiques d'un protocole en mode orienté connexion.
2. Explicitez les mécanismes nécessaires à la réalisation d'un transfert de données fiable.
3. Indiquez les principales caractéristiques du protocole TCP.

1.2.1 Gestion de connexion

1. Comment sont différenciés les rôles des segments dans le cadre de TCP ?
2. Représentez le diagramme d'établissement d'une connexion réseau. Discutez du nombre de messages nécessaires. Dans le contexte de TCP, pourquoi procéder à un échange en trois phases ?
3. Quelles sont les possibilités de numérotation des segments ? Dans le contexte de TCP, comment évoluent les numéros de séquence ? Deux segments successifs peuvent-ils contenir le même numéro de séquence ? Et en l'absence de données transmises, le numéro de séquence peut-il augmenter ?
4. Pourquoi ne pas commencer la numérotation de séquence à 0 ?
5. Quelles sont les possibilités de terminaison d'une connexion ? Représentez les diagrammes correspondants.

1.2.2 Gestion de la fiabilité

1. La fiabilisation requiert la connaissance de la remise des données. Quelles sont les deux principales techniques d'acquittement ? Précisez leurs intérêts respectifs selon l'importance du trafic contrôlé et précisez laquelle est utilisée par TCP.
2. En cas de perte de données, deux politiques de retransmission sont envisageables : décrivez-les et indiquez celle utilisée avec TCP.

1.2.3 Estimation du RTT d'une connexion

Lorsque l'on utilise le protocole TCP, le choix du RTT (*Round Trip Time*) est important puisque la détection de perte en découle directement et que les divers mécanismes de contrôle qui vont influer sur le débit d'émission en dépendent. Le calcul du RTT peut se faire à l'aide de la formule suivante $RTT = \alpha * RTT_{mesure} + (1 - \alpha) * RTT_{ancien}$ avec α le coefficient de lissage.

1. Comment TCP mesure-t-il le délai aller-retour (RTT_{mesure}) pour un segment donné ?
2. Montrez que l'effet d'une valeur mesurée pour le RTT se réduit exponentiellement avec le temps.
3. Quel est l'intérêt d'utiliser cette formule comparée à une moyenne mobile dans laquelle le RTT est la moyenne calculée sur une fenêtre de longueur L ?
4. Quelles sont les conséquences d'une valeur de α proche de 1 ou proche de 0 ?
5. Quelles sont les précautions à prendre lors de la mesure du délai aller-retour d'un segment donné ?
6. A votre avis, quelle est l'utilité de l'option TCP `timestamp` ? Pourquoi est-il conseillé d'utiliser cette option (on pourra consulter le RFC 1323 pour plus de détails) ?

1.2.4 Calcul du RTO de TCP

1. La première approche pour déterminer la valeur du temporisateur de retransmission RTO (*Retransmission TimeOut*) est $RTO = n * RTT$. Quelles sont les précautions à prendre quant au dimensionnement de n ?
2. La deuxième approche utilise $RTO = RTT + \delta D$ avec généralement $\delta = 4$.
   $D = \beta(|RTT_{mesure} - RTT_{ancien}|) + (1 - \beta)D_{ancien}$ avec généralement $\beta = 1/4$.
   Cette approche consiste à calculer la variance du RTT. Quelle est l'amélioration apportée ?
3. Comment calculer le RTO lorsqu'il y a des pertes ?

2 Observation de trafic UDP

Les analyses qui suivent ont pour but d'observer les mécanismes protocolaires du protocole UDP.

2.1 Caractéristiques d'un datagramme UDP (sans machine)

Voici la trace d'une trame à étudier :

```
0000  00 04 76 21 1b 95 00 01  02 a5 fb 88 08 00 45 00   ..v!.... ......E.
0010  00 30 00 00 40 00 40 11  6d 58 c2 fe a3 b1 c2 fe   .0..@.@. mX......
0020  a3 b6 06 9c 00 45 00 1c  e1 e4 00 01 75 6e 69 78   .....E.. ....unix
0030  62 6f 74 74 00 6e 65 74  61 73 63 69 69 00         bott.net ascii.
```

1. Analysez **manuellement** (sans `wireshark`) la trame présentée ci-dessus. Utilisez directement le support du lab pour entourer les différents champs sur les traces.

2. Quelles informations peut-on déduire des numéro de ports contenus dans le datagramme ci-dessus ?
3. UDP est dit minimaliste en terme de fonctionnalités. En observant les champs présents dans l'en-tête, pensez-vous que leur nombre soit réduit au maximum ?

2.2 Capture et analyse de datagrammes UDP

2.2.1 Capture d'un trafic UDP

Cette première capture a pour but de percevoir la nature du trafic UDP. Sur la plateforme d'expérimentation, la topologie 1 a été configurée (postes clients et serveur sur le même LAN). Générez du trafic UDP en utilisant TFTP. Réalisez la capture de ce trafic à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles
  - fenêtre 1 (VM "client") : tapez `ssh -X etudiant@10.0.7.`**N**1
  - fenêtre 2 (VM "sonde") : tapez `ssh -X root@10.0.7.`**N**2 (attention, vous êtes administrateur)
  - fenêtre 3 (VM "serveur") : tapez `ssh -X etudiant@10.0.7.`**N**3
- Vérifiez que le serveur TFTP tourne sur `10.0.7.`**N**3 (fenêtre 3)
  - recherchez le processus du serveur, (tapez `ps aux | grep tftp` ou vérifiez dans `inetd.conf`)
  - verifiez qu'un répertoire est configuré pour réaliser les transferts
  - visualisez la configuration des interfaces, en particulier celle sur le LAN d'étude (`/sbin/ifconfig eth1`) pour vérifier l'adresse IPv4 du serveur pour la connexion du client (devrait être `10.`**N**`.1.`**N**3)
- Démarrez la capture en lançant l'analyseur sur `10.0.7.`**N**2 (fenêtre 2)
  - lancez l'analyseur, tappez : `wireshark`
  - initiez la capture sur l'interface `eth1`, comme indiqué dans le **Lab n°1**
- Démarrez un client TFTP sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en établissant une connexion vers le serveur, tapez : `10.`**N**`.1.`**N**3
  - envoyez un fichier au serveur avec la commande `put`
  - terminez l'échange avec la commande `quit`
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à UDP** (filtre = `udp`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

2.2.2 Analyse de l'échange UDP

1. Par rapport à la capture des échanges TFTP du **Lab n°2** (avec filtre = `tftp`), quelles différences observez-vous avec la trace **affichée** ici ?
2. Pouvez-vous identifier les rôles de client ou de serveur des applications impliquées ?
3. Comment est gérée l'association des deux applications impliquées ?
4. UDP n'intégrant pas de mécanismes de fiabilité, que pouvez-vous dire des mécanismes de protection mis en place par les applications ?

2.2.3 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme4-udp.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

3 Observation du trafic TCP

Cette seconde analyse a pour but d'observer les différents mécanismes protocolaires de TCP. Pour cela, nous nous appuierons sur des captures de segments TCP.

3.1 Mise en place de la connexion TCP (sans machine)

Voici la première trame échangée lors de l'ouverture d'une connexion :

```
0000  00 50 7f 05 7d 40 00 10  a4 86 2d 0b 08 00 45 00   .P..}@.. ..-...E.
0010  00 3c 17 96 40 00 40 06  6d f3 0a 21 b6 b2 c0 37   .<..@.@. m..!...7
0020  34 28 84 b3 00 50 b6 94  b0 b7 00 00 00 00 a0 02   4(...P.. ........
0030  16 d0 e8 23 00 00 02 04  05 b4 04 02 08 0a 00 6f   ...#.... .......o
0040  a7 21 00 00 00 00 01 03  03 00                     .!...... ..
```

1. Analysez **manuellement** (sans `wireshark`) la trame présentée ci-dessus. Utilisez directement le support du Lab pour entourer les différents champs sur les traces.
2. Quels sont les bits de contrôle (TCP *flags*) positionnés ? Que signifient-ils ?
3. Identifiez les hôtes impliqués dans cet échange. Quels vont être leurs rôles respectifs dans la suite ?
4. Quelles informations peut-on déduire des numéro de ports contenus dans les segments ci-dessus ?
5. Rappelez le fonctionnement des numéros de séquence de TCP. Justifiez les valeurs présentes dans ce segment.
6. Pouvez-vous observer des options dans l'en-tête TCP ? Si oui, que signifient-elles ?

3.2 Capture et analyse d'une connexion TCP

3.2.1 Capture d'un trafic TCP

Cette seconde capture a pour but de percevoir la nature du trafic TCP. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic HTTP à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles (si ce n'est déjà fait).
- Vérifiez que le serveur HTTP (`apache2`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrer la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Démarrez un client HTTP sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client de votre choix (`firefox`...)
  - établissez une connexion vers le serveur en spécifiant l'URL suivante : `http://10.`**N**`.1.`**N**3 (affiche la page par défaut du serveur Apache)
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à TCP** (filtre = `tcp`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

3.2.2 Analyse de l'échange TCP

1. Par rapport à la capture des échanges HTTP du **Lab n°2** (avec filtre = `http`), quelles différences observez-vous avec la trace **affichée** ici ?
2. Quels sont les bits de contrôle (TCP *flags*) positionnés dans les différentes trames ? Que signifient-ils ?
3. Vérifiez le fonctionnement des numéros de séquences de TCP. **Attention,** `wireshark` **utilise une numérotation relative.** Comparez les valeurs réelles lues dans la trame et celles données par `wireshark` pour les numéros de séquence et d'acquittement. Justifiez les valeurs présentés.
4. Que pouvez-vous dire du contrôle de flux pour les segments étudiés ?
5. Pouvez-vous trouvez d'autres options dans les entêtes TCP ? Si oui, que signifient-elles ?

3.2.3 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme4-tc1.dmp` (similaire à celle capturée précédemment) puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

3.3 Analyse détaillée d'un échange TCP

Utilisez la trace de sauvegarde `tme4-tc1.dmp` proposée ci-dessus, puis analysez-la sur le poste ARI avec le logiciel `wireshark` exécuté sans les droits d'administrateur.

Ci-dessous sont affichées les premières trames de cette trace partiellement décodée grâce à l'outil UNIX `tcpdump` (basé sur `libpcap`, la même bibliothèque de capture que `wireshark`, mais plus adaptée pour une présentation textuelle) :

```
00:00:00.000000 IP 10.33.182.178.33971 > 192.55.52.40.80: Flags [S], seq 3063197879, win 5840,
[A]                                    options [mss 1460,sackOK,TS val 7317281 ecr 0,nop,wscale 0], length 0

00:00:00.170558 IP 192.55.52.40.80 > 10.33.182.178.33971: Flags [S.], seq 610765288, ack 3063197880, win 64240,
[B]                                    options [mss 1402,nop,wscale 0,nop,nop,TS val 0 ecr 0,nop,nop,sackOK], length 0

00:00:00.170618 IP 10.33.182.178.33971 > 192.55.52.40.80: Flags [.], ack 1, win 5840,
[C]                                    options [nop,nop,TS val 7317298 ecr 0], length 0

00:00:00.170819 IP 10.33.182.178.33971 > 192.55.52.40.80: Flags [P.], seq 1:486, ack 1, win 5840,
[D]                                    options [nop,nop,TS val 7317298 ecr 0], length 485

00:00:00.370505 IP 192.55.52.40.80 > 10.33.182.178.33971: Flags [.], seq 1:1391, ack 486, win 63755,
[E]                                    options [nop,nop,TS val 19332362 ecr 7317298], length 1390

00:00:00.370560 IP 10.33.182.178.33971 > 192.55.52.40.80: Flags [.], ack 1391, win 8340,
[F]                                    options [nop,nop,TS val 7317318 ecr 19332362], length 0

00:00:00.381289 IP 192.55.52.40.80 > 10.33.182.178.33971: Flags [.], seq 1391:2781, ack 486, win 63755,
[G]                                    options [nop,nop,TS val 19332362 ecr 7317298], length 1390

00:00:00.381336 IP 10.33.182.178.33971 > 192.55.52.40.80: Flags [.], ack 2781, win 11120,
[H]                                    options [nop,nop,TS val 7317319 ecr 19332362], length 0

...
```

1. Tracez **précisément** le chronogramme correspondant à cet échange **(Respectez impérativement l'échelle des temps pour réussir à visualiser l'évolution des échanges)**.
2. Nous souhaitons étudier la demi-connexion correspondant à l'émission de données du serveur (`192.55.52.40.www`) vers le client (`10.33.182.178.33971`). Complétez les dernières lignes du tableau suivant (numéros de séquence relatifs) :

action	base de la fenêtre	pointeur d'émission	fin de la fenêtre	taille de la fenêtre	*commentaire*
réception A	—	—	—	5840	*win 5840*
émission B	0 (610765288)	1	5840	—	*SYN, +1 pas d'émission*
réception C	1	1	5841	5840	*ACK*
réception D	1	1	5841	5840	*ACK*
émission E	1	1391	5841	—	
...					
...					
...					
...					

3. Commentez l'évolution des numéros de séquence.
4. Que pouvez-vous dire sur la gestion des tampons ?
5. Observez-vous de nouvelles options ? Pouvez-vous les expliquer ?
6. Que pouvez-vous dire à propos de la génération des acquittements par le récepteur ?
7. Comment se termine la communication ? Détaillez les échanges finaux.

4 Echanges TCP imbriqués

4.1 Capture et analyse d'un échange avec deux connexions TCP imbriquées

4.1.1 Capture du trafic de deux connexions TCP imbriquées

Cette dernière capture a pour but de percevoir l'entrelacement des deux connexions TCP associées à l'application FTP. Sur la plateforme d'expérimentation, toujours sur la topologie 1 (postes client et serveur sur le même LAN), réalisez la capture de trafic FTP à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles.
- Vérifiez que le serveur FTP (`ftpd`) tourne sur `10.0.7.`**N**3 (fenêtre 3)
- Démarrez la capture en lançant l'analyseur sur sur l'interface `eth1` de l'hôte `10.0.7.`**N**2 (fenêtre 2)
- Démarrez un client FTP sur `10.0.7.`**N**1 (fenêtre 1)
  - lancez le client en établissant une connexion vers le serveur, tapez : `ftp 10.`**N**`.1.`**N**3 (réseau d'expérimentation)
  - identifiez-vous avec le login `etudiant` et le mot de passe correspondant
  - choisissez un fichier et transférez le sur la machine client (commandes de l'interface utilisateur `get`)
  - terminez l'échange (commandes de l'interface utilisateur `quit`)
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à TCP** (filtre = `tcp`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application afin de pouvoir démarrer les analyses qui suivent.

4.1.2 Analyse du trafic de deux connexions TCP imbriquées

Retrouvez la correspondance entre les messages échangés sur le réseau, sur la connexion de contrôle de FTP et ceux affichés par l'application (interface utilisateur sur le client).

1. Observez l'échange capturé et expliquez les actions réalisées au niveau applicatif.
2. Tracez le chronogramme correspondant à ces échanges en utilisant une couleur différente par connexion.
3. Que pouvez-vous dire de l'utilisation du *flag* `PUSH` ?

4.1.3 Sans la plateforme...

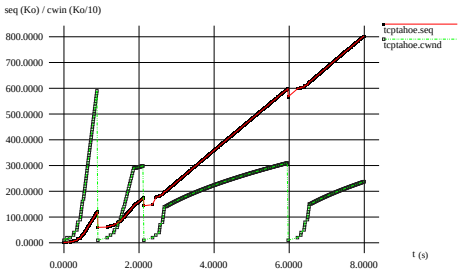
En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme4-tc2.dmp` (similaire à celle capturée précédemment) puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

5 Avant de quitter la salle

- Si vous avez enregistré des captures sur la VM "sonde", n'oubliez pas de les rapatrier sur votre compte utilisateur de l'ARI. Tapez : `scp root@10.0.7.N2:<ma_trace> <destination_locale>`.
- Avant de terminer vos connexions sur les équipements de la plateforme, supprimez vos fichiers et modifications effectuées afin de retrouver l'état initial.

ARES - Lab n°5

Couche transport (2) : Contrôle de congestion TCP



1 Contrôle de congestion (sans machines)

TCP est utilisé pour le transport fiable de données dans l'Internet. Nous avons précédemment étudié la gestion des connexions et les mécanismes TCP. Dans les exercices suivants, nous allons nous intéresser à un autre comportement fondamental de TCP : le contrôle de congestion.

1.1 Détection de la congestion

La conception de TCP date de la fin des années 70. Plusieurs algorithmes de contrôle de congestion ont été intégrés depuis, principalement suite aux travaux de Van Jacobson publiés en 1988. Ces derniers continuent à évoluer dans les différentes variantes de TCP. Les exercices proposés dans la suite sont fondés sur les dernières mises à jour : le RFC 5681 de septembre 2009.

1. Pour TCP, quel phénomène indique une congestion dans le réseau ?
2. Que se passe-t-il dans un routeur pour susciter ce phénomène ?
3. Pour TCP, ce phénomène permet de déduire la congestion. Mais celui-ci peut aussi se produire quand il n'y a pas de congestion dans le réseau. Dans quels autres cas un tel phénomène peut-il apparaître ?
4. Si ce phénomène n'indique pas toujours une congestion, pourquoi TCP se base-t-il sur cette inférence ? Pourquoi n'utilise-t-on pas une approche où le routeur constatant la congestion envoie un message explicite à l'émetteur ?

1.2 Algorithmes de contrôle de congestion

Pour le contrôle de congestion, TCP utilise un seuil qui indique le débit au-dessus duquel la congestion risque de se produire. Ce seuil est exprimé par le paramètre `ssthresh` (en octets). Pour obtenir le débit seuil on divise `ssthresh` par le RTT (*Round Trip Time*). Le débit peut varier en-dessous et au-dessus du seuil `ssthresh`$/RTT$. L'émetteur maintient un second paramètre, `cwnd` (taille de la fenêtre de congestion) qui indique le nombre maximum d'octets qu'il peut envoyer avant de recevoir un acquittement. Quand `cwnd > ssthresh`, l'émetteur fait particulièrement attention à ne pas provoquer de congestion.

1. Supposons que `ssthresh` soit à 5000 octets, `cwnd` est à 6000 octets, et la taille d'un segment est de 500 octets. Un émetteur envoie douze segments de 500 octets dans une période `RTT`, et reçoit douze acquittements (un pour chaque segment). Que deviennent les valeurs de `ssthresh` et `cwnd` ? Comment s'appellent ces changements de valeurs ?
2. Supposons que `ssthresh` soit toujours à 5000 octets, que `cwnd` est maintenant à 14.000 octets, que l'émetteur envoie $14.000/500 = 28$ segments, et que l'émetteur reçoive une indication de congestion avant de recevoir le premier acquittement. Que deviennent les valeurs de `ssthresh` et `cwnd` ? Comment s'appellent ces changements de valeurs ?
3. Nous venons de voir comment augmente et diminue `cwnd` en fonction de l'absence ou la présence d'indicateurs de congestion. Comment s'appelle cet algorithme ? Sur quel principe repose cet algorithme ?
4. Au démarrage, ou après avoir reçu une indication de congestion, la valeur de `cwnd` est plus petite que la valeur de `ssthresh`. Décrivez la manière permettant d'augmenter `cwnd` quand celle-ci est inférieure à `ssthresh`, en fonction de l'exemple suivant. Supposons que `ssthresh` soit égal à 3000 octets et que `cwnd` soit égal à 500 octets, la taille d'un segment. L'émetteur a plusieurs segments prêts à être envoyés. Combien de segments envoie l'émetteur pendant la première

période RTT ? S'il reçoit des acquittements pour tous ses segments, que devient la valeur de `cwnd` ? Combien de segments envoie l'émetteur pendant la deuxième période RTT ? S'il reçoit des acquittements pour tous ses segments, que devient la valeur de `cwnd` ? En général, comment évolue la taille de `cwnd` ?

5. Comment s'appelle la période pendant laquelle `cwnd` est plus petit que `ssthresh` ?
6. Que devient la valeur de `ssthresh` si l'émetteur reçoit une indication de congestion pendant que `cwnd` est plus petit que `ssthresh` ?

1.3 Débit moyen d'une connexion TCP

Supposons que nous souhaitions effectuer un transfert de données de taille importante à travers une connexion TCP.

1. En négligeant la période pendant laquelle `cwnd` est plus petit que `ssthresh`, montrez que le débit moyen d associé à une connexion TCP est égal à :

$$d = \frac{3}{4}\frac{W * MSS}{RTT}$$

où W est la taille de la fenêtre (en segment) au moment de la congestion, MSS la taille d'un segment (supposée maximale), et RTT est le délai aller-retour (supposé constant durant la période de la transmission).

2. Montrez que le taux de pertes p est égal à :

$$p = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

3. Montrez que si le taux de pertes observé par une connexion TCP est p, alors son débit moyen d peut être approximé par :

$$d = \frac{1,22 * MSS}{RTT\sqrt{p}}$$

4. Quels autres paramètres peuvent influer sur le débit d'une connexion TCP ?
5. Quelle utilité voyez-vous à la relation calculée dans la dernière formule de d ?

2 Etude de la latence d'un serveur web (sans machine)

Nous souhaitons étudier la latence liée à la réponse à une requête HTTP[1]. Nous faisons les hypothèses simplificatrices suivantes :

- Le réseau n'est pas congestionné (pas de pertes ni de retransmissions) ;
- Le récepteur est doté de tampons de réception infinis (limitation de l'émetteur uniquement liée à la fenêtre de congestion) ;
- La taille de l'objet à recevoir du serveur est O, un multiple entier du MSS (MSS à pour taille S bits) ;
- Le débit de la liaison connectant le client au serveur est R (bits/s) et on néglige la taille de tous les entêtes (TCP, IP et liaison). Seuls les segments transportant des données ont un temps de transmission significatif. Le temps de transmission des segments de contrôle (ACK, SYN...) est négligeable ;
- La valeur du seuil initial du contrôle de congestion n'est jamais atteinte ;
- La valeur du délai aller-retour est RTT.

1. Dans un premier temps, nous supposons que nous n'avons pas de fenêtre de contrôle de congestion. Dans ce cas montrez que la latence L peut s'exprimer de la manière suivante :

$$L = 2RTT + O/R$$

[1] Latence d'une requête HTTP : laps de temps pour la création de la connexion et la récupération de l'intégralité de l'objet demandé.

2. Nous supposons maintenant une fenêtre de congestion **statique** de taille fixe égale à W. Calculez la latence dans ce premier cas :

$$WS/R < RTT + S/R$$

3. Nous supposons toujours une fenêtre de congestion **statique** de taille fixe égale à W. Calculez la latence dans ce second cas :

$$WS/R > RTT + S/R$$

4. Comparez la latence obtenue avec une fenêtre de contrôle de congestion **dynamique** (*slow-start*) avec celle sans contrôle de congestion.

5. Application numérique :

R	O/R	L (sans *slow start*)	K'	L (latence globale de TCP)
56 Kbps				
512 Kbps				
8 Mbps				
100 Mbps				

K' est le nombre de fenêtres envoyées avant de passer au second cas ($log2(1 + RTT * R/S)$). Considérez trois cas :

(a) S= 512 octets, RTT= 100 msec, O=**100 Koctets** (=200S) ;

(b) S= 512 octets, RTT= 100 msec, O=**5 Koctets** (=10S) ;

(c) S= 512 octets, RTT= **1 seconde**, O=5 Koctets (=10S).

3 Analyse des mécanismes TCP

Voici quatre captures de trafic HTTP. Les trois premières sont pré-enregistrées et reposent sur des clients et serveurs éloignés (les trois sont réalisées avec une sonde proche du client). Vous réaliserez la dernière capture dans un environnement local (sur la plate-forme d'expérimentation). Pour chacune d'elles, tracez précisément le chronogramme et étudiez les mécanismes de contrôle de congestion mis-en-œuvre. Discutez en particulier des points suivants :

1. Quel est le RTT moyen ?
2. Reconnaissez-vous les mécanismes de contrôle de congestion ?
3. Jusqu'à combien de segments sont transmis par RTT ?
4. Quel est le débit moyen alors atteint ?
5. Un envoi continu apparaît-il ?
6. Des perturbations sont-elles présentes (déséquencement, retransmission...) ?

3.1 Trafic HTTP Paris-Brisbane (WAN Intercontinental : 17000km)

Utilisez le logiciel `wireshark` (sans avoir besoin des droits d'administrateur) sur le poste ARI pour cette première trace. Télécharger la trace `tme5-wau.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`.

3.2 Trafic HTTP Paris-Budapest (WAN Continental : 1200km)

Utilisez le logiciel `wireshark` (sans avoir besoin des droits d'administrateur) sur le poste ARI pour cette trace. Télécharger la trace `tme5-whu.dmp` (similaire à celle capturée précédemment) des emplacement utilisés précédemnets

3.3 Trafic HTTP Paris-Evry (MAN : 36km)

Utilisez le logiciel `wireshark` (sans avoir besoin des droits d'administrateur) sur le poste ARI pour cette trace. Télécharger la dernière trace `tme5-man.dmp` (similaire à celle capturée précédemment) des emplacement utilisés précédemnets

3.4 Trafic HTTP local (LAN)

3.4.1 Capture du trafic TCP résultant d'un échange HTTP local

Cette première capture a pour but de percevoir la nature du trafic TCP dans un LAN. Sur la plateforme d'expérimentation, la topologie 1 a été configurée (postes clients et serveur sur le même LAN). Réalisez la capture de trafic TELNET à l'aide du logiciel `wireshark` :

- A partir du poste ARI **N**, connectez-vous sur les 3 VM correspondantes de la plateforme à l'aide de 3 fenêtres textuelles
  - fenêtre 1 (hôte "client") : tapez `ssh -X etudiant@10.0.7.`**N**1
  - fenêtre 2 (hôte "sonde") : tapez `ssh -X root@10.0.7.`**N**2 (attention, vous êtes administrateur)
  - fenêtre 3 (hôte "serveur") : tapez `ssh -X etudiant@10.0.7.`**N**3
- Vérifiez qu'un gros fichier est accessible depuis le répertoire `public_html` du compte `etudiant` (à créer avec les droit d'accès publics si absent) et que le serveur HTTP tourne sur la VM `10.0.7.`**N**3 (fenêtre 3)
  - générez un gros fichier, tapez : `dd if=/dev/zero of=public_html/fichier100Mo bs=1M count=100`
  - vérifiez l'accès public du fichier(`-rw-r--r--`), taper : `ls -l public_html/fichier100Mo`
  - vérifiez que le serveur HTTP tourne (`apache2`)
  - visualisez la configuration des interfaces, en particulier celle sur le LAN d'étude pour vérifier l'adresse IP du serveur pour la connexion du client (devrait être `10.`**N**`.1.`**N**3)
- Démarrez la capture en lançant l'analyseur sur `10.0.7.`**N**2 (fenêtre 2)
  - lancez l'analyseur, tappez : `wireshark`
  - initier la capture sur l'interface `eth1`, comme indiqué précédemment
- Démarrez un client HTTP sur `10.0.7.`**N**1 (fenêtre 1)
  - démarrez le client HTTP de votre choix (`firefox`...)
  - tapez l'URL suivante `http://10.`**N**`.1.`**N**`3/~etudiant/fichier100Mo`
- Observez la capture se réaliser dans la fenêtre de `wireshark`
- Terminez la capture. **Filtrez le trafic afin de ne conserver que celui relatif à TCP** (filtre = `tcp`). Enregistrez la trace filtrée pour pouvoir la ré-utiliser ultérieurement. Ne quittez pas l'application et répondez aux mêmes questions que pour les trois traces pré-enregistrées précédentes.

3.4.2 Sans la plateforme...

En cas de problème d'accès à la plateforme d'expérimentation ou si vous souhaitez réviser sur une autre machine, vous pouvez télécharger la trace `tme5-lan.dmp` (similaire à celle capturée précédemment) soit à partir du répertoire `/Infos/lmd/2013/master/ue/ares-2013oct`, soit sur la page web `http://www-rp.lip6.fr/~fourmaux/Traces/labV6.html`, puis l'analyser avec le logiciel `wireshark` (sans avoir besoin des droits d'administrateur).

4 Avant de quitter la salle

- Détruisez le gros fichier (`rm public_html/fichier100Mo`).
- Si vous avez enregistré des captures sur la VM "sonde", n'oubliez pas de les rapatrier sur votre compte utilisateur de l'ARI. Tapez : `scp root@10.0.7.`**N**`2:<ma_trace> <destination_locale>`.
- Avant de terminer vos connexions sur les équipements de la plateforme, supprimez vos fichiers et modifications effectuées afin de retrouver l'état initial.