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'UE 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 IPv4** : décimale pointée (ex : `10.1.1.3`)
  - **Adresses IPv6** : hexadécimale double pointée compacte (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  .. ..
```

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.

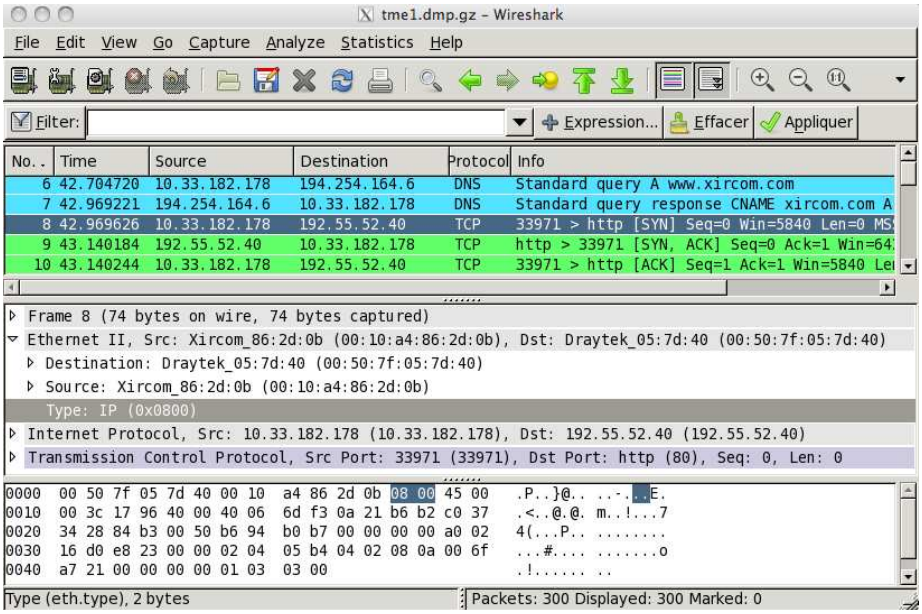


FIGURE 1 : Fenêtre principale de `wireshark`

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-npa.lip6.fr/~fourmaux/Traces/labV8.html`

2.1 Introduction à `wireshark`

Sur la machine face à votre binôme, connectez-vous à votre compte sous GNU/Linux. Sur cet ordinateur, géré par la PPTI[2] et qui accède à l'Internet et diverses ressources sensibles, vous n'avez pas les droits d'administrateur. Les logiciels d'analyse de trames requièrent ceux-ci pour réaliser des captures à partir de votre carte réseau. Mais, vous pouvez utiliser – avec des droits limités – leur fonctionnalité d'analyse multi-protocolaire sur les machines de la PPTI. Dans les sections suivantes nous étudierons comment réaliser des captures... mais sur d'autres machines.

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/2019/master/ue/MU4IN001-2019oct/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] `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 **x86** avec système **GNU/Linux** que vous utilisez actuellement). Vous pouvez le télécharger sur `http://www.wireshark.org`.

[2] PPTI : Plateforme Pédagogique et Technique d'Informatique

[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`.

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 PPTI, 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.

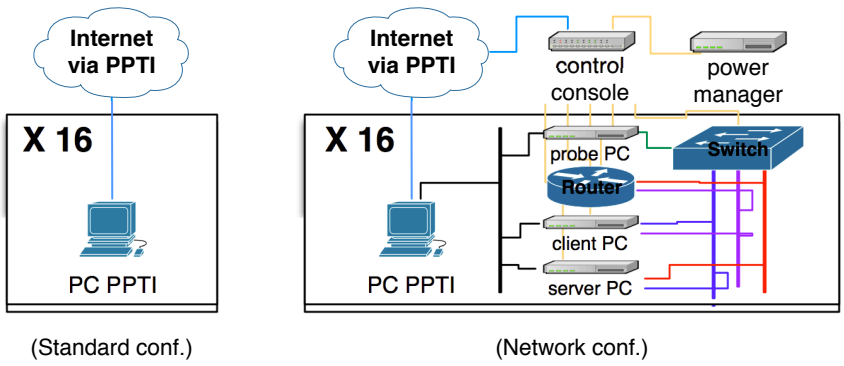
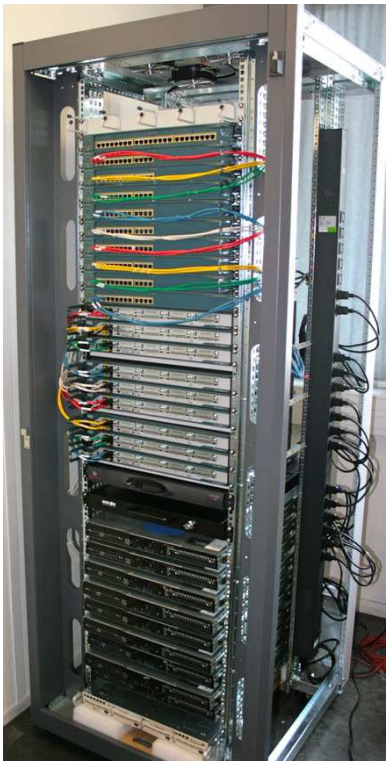


FIGURE 2 : Evolution du poste étudiant

L'ajout de cette plateforme réseau permet de faire évoluer le poste PPTI 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 la PPTI (accès aux comptes utilisateurs, aux logiciels habituels et à l'Internet). Cette évolution est présentée dans la FIGURE 2.

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 en rack 1U**, Intel Xeon E3-1230v5, 16 Go RAM, HD 1 To, 4 NIC Ethernet 1GBaseT, 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 la PPTI de la salle. Ces postes sont des PC équipés de deux cartes réseau. L'une permet l'accès au réseau habituel de la PPTI 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 PPTI de cette salle (31-208) ou à distance via SSH sur ce même poste (`ssh -Y ppti-14-503-`**N**). Il n'y a pas de routage ou relayage entre le réseau de la PPTI et celui de la plateforme assurant ainsi l'isolation du réseau expérimentation.

Les postes nommés `ppti-14-503-01` à `ppti-14-503-08` sont connectés sur la baie 1 et ceux nommés `ppti-14-503-09` à `ppti-14-503-16` sur la baie 2.

Appelons **N** la valeur du dernier nombre du nom de la machine PPTI utilisée. Le poste PPTI **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 viruelles (VM) :
  - la VM "client" **N**1
  - la VM "sonde" **N**2
  - la VM "serveur" **N**3

La configuration des VM est présentée dans la FIGURE 3.

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

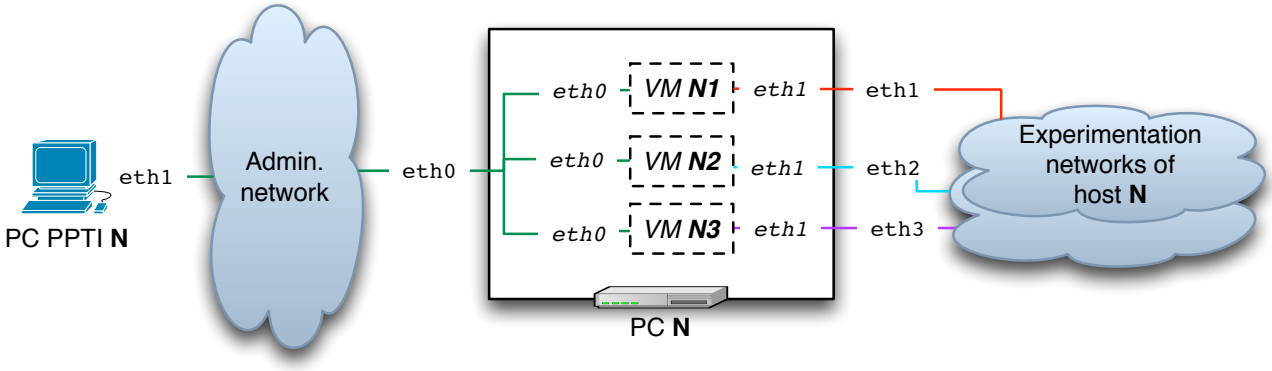


FIGURE 3 : Configuration des 3 VM de la plate-forme

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 PPTI 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 la PPTI 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.

3.3.3 Topologie 3 (avec plusieurs routeurs – labs futures)

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.

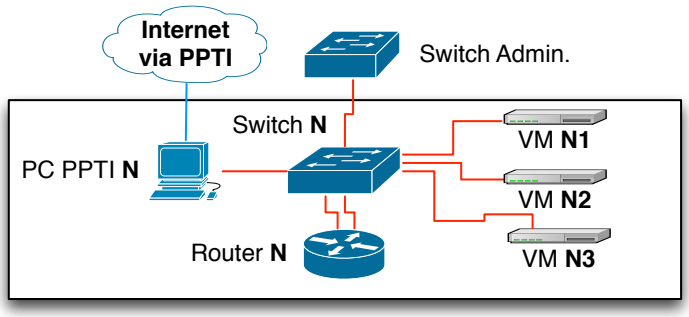


FIGURE 4 : Topologie physique associée à un poste PPTI

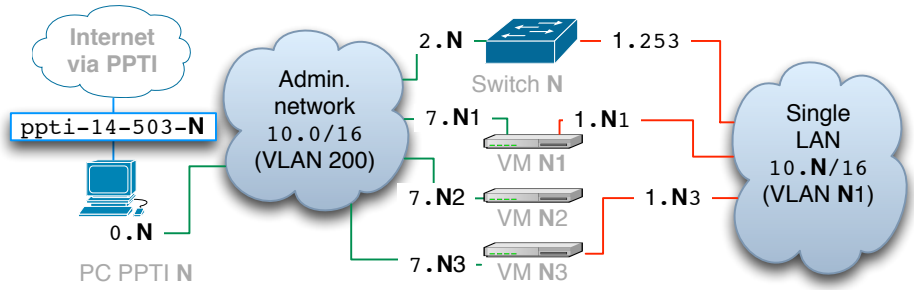


FIGURE 5 : Topologie virtuelle 1 (un LAN)

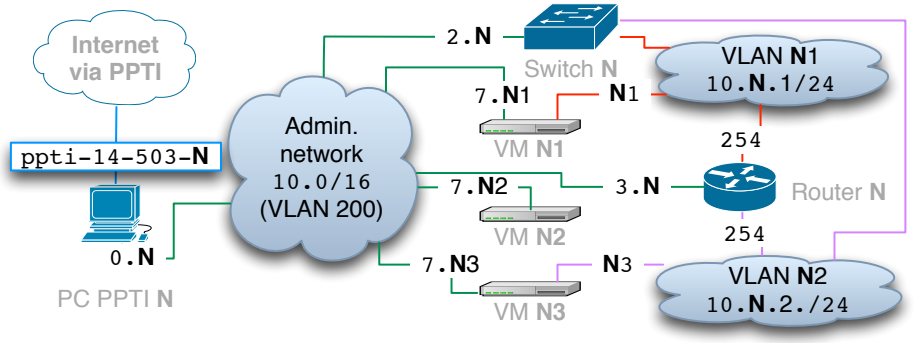


FIGURE 6 : Topologie virtuelle 2 (deux LAN et un routeur)

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 d'accès du poste PPTI **N** via le réseau d'administration : `10.0.0.`**N**
  - les commutateurs : `10.0.2.`**N**
  - les routeurs : `10.0.3.`**N**
  - les VM "client" **N**1 (`eth0`) : `10.0.7.`**N**1
  - les VM "sonde" **N**2 (`eth0`) : `10.0.7.`**N**2
  - les VM "serveur" **N**3 (`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 PPTI est relié au réseau d'administration de la plate-forme d'expérimentation par une interface locale (voir sur la FIGURE 3). La commande Unix `/sbin/ifconfig` permet de vérifier la configuration des interfaces d'une machine et connaitre leur nom. Si l'interface avec l'adresse IPv4 `10.0.0.`**N** est inexistante, relancer-la ou redémarrez votre poste PPTI.

3.4.1 Contrôle à distance des 3 VM de la plateforme via SSH et tunnel X11

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

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

L'option "-Y" 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.

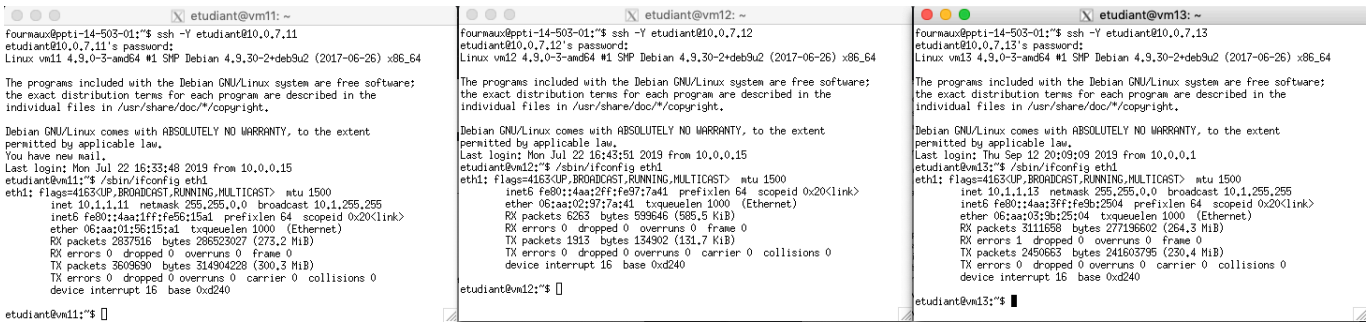


FIGURE 7 : Sessions SSH à partir du poste `ppti-14-503-01`

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.

Dans la suite, tous les Labs seront présentés avec cette technique d'accès à distance : SSH a pour avantage d'être standard, sécurisée et utilisable à longue distance.

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.

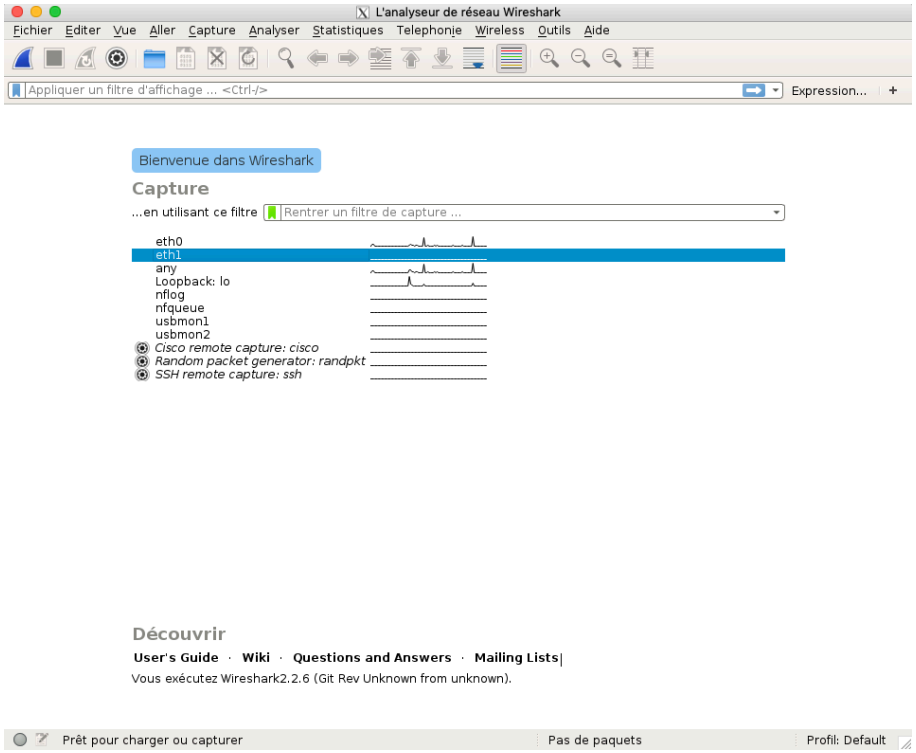


FIGURE 8 : Démarrage de `wireshark`

Lancez `wireshark` de la fenêtre de la "sonde" et une nouvelle fenêtre en provenance de la VM "sonde" apparaîtra. 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).

Si une nouvelle fenêtre apparaît, 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 10).

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.

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 PPTI permet de se

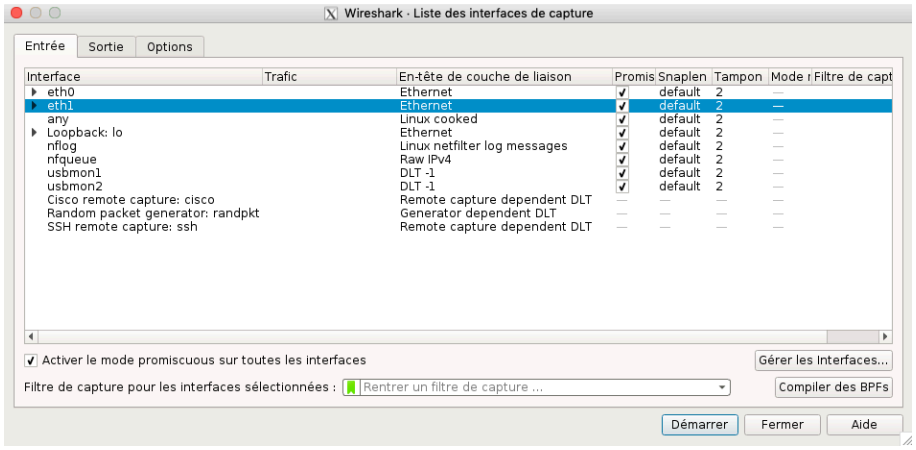


FIGURE 9 : Démarrage d'une capture à partir de la VM "sonde" : liste des interfaces

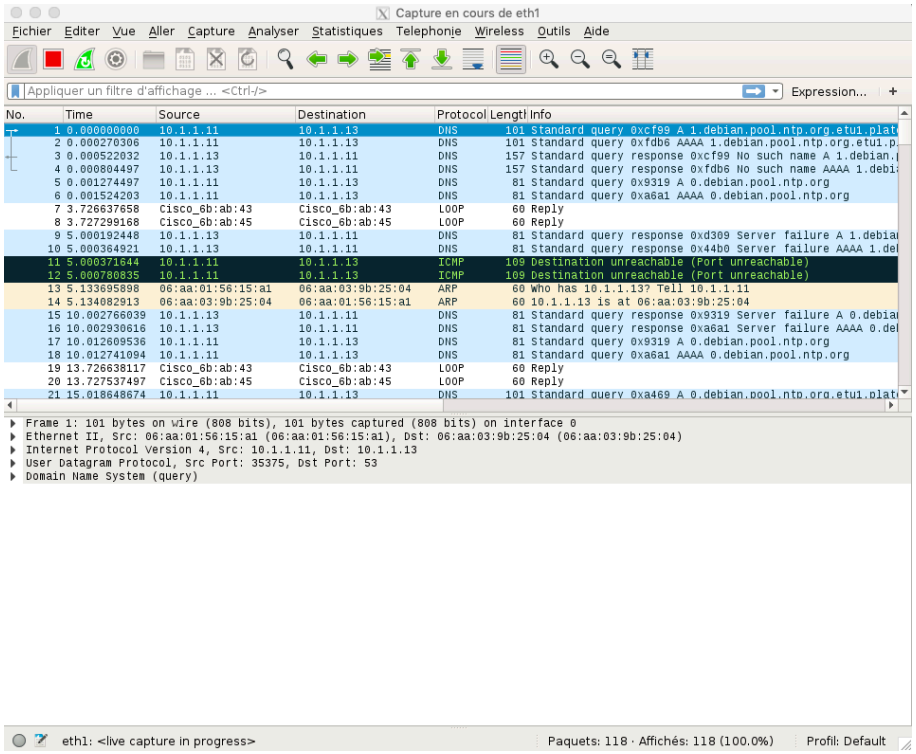


FIGURE 10 : Capture de trafic `ping` (paquets ICMP)

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, vm**N**1 (hôte "client") : tapez `ssh -Y etudiant@10.0.7.`**N**1
  - fenêtre 2, vm**N**2 (hôte "sonde") : tapez `ssh -Y etudiant@10.0.7.`**N**2
  - fenêtre 3, vm**N**3 (hôte "serveur") : tapez `ssh -Y 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 11. N'oubliez pas de terminer la capture.

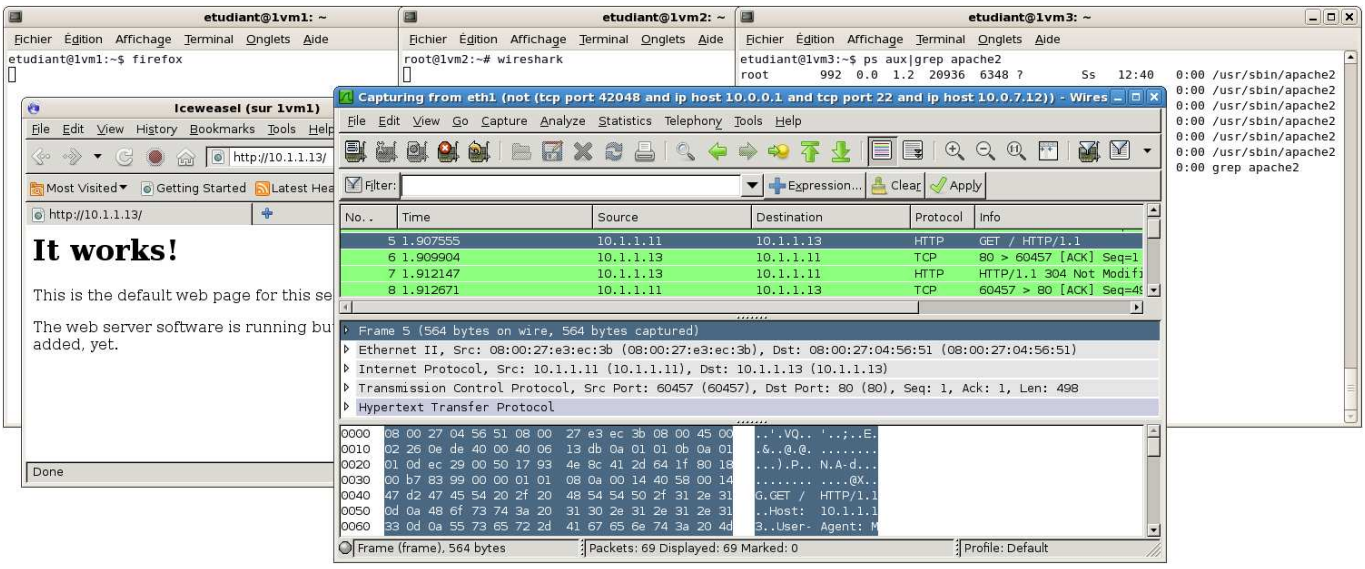


FIGURE 11 : Capture de trafic HTTP

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