U.F.R. Sciences de l'Homme et de la Société
Master MIASHS

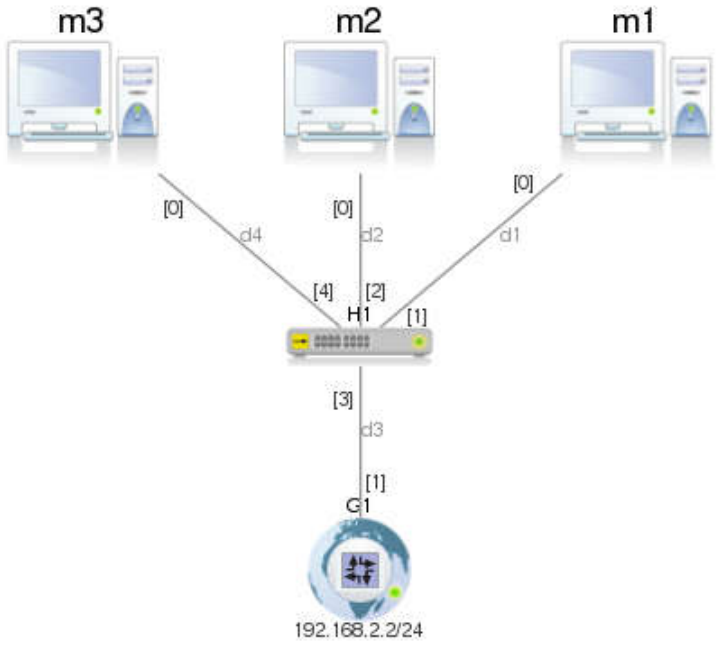
TP 3 sur IP

L'objectif de ce troisième TP est de vous faire comprendre :

- le fonctionnement du protocole DHCP
- le fonctionnement du protocole de transport TCP à travers le protocole applicatif HTTP

Configuration de base

Téléchargez le projet Marionnet `capture3.mar`. Démarrez l'ensemble des composants.



A l'aide de la commande `ifconfig` attribuez à la machine *m1,* l'adresse IP 192.168.2.254
Rappel de la syntaxe de la commande :
`ifconfig eth0` *`adresse_ip`* **`netmask`** *`masque_reseau`* **`up`**

Analyse du protocole DHCP
(*Dynamic Host Configuration Protocol*)

DHCP est un protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres IP d'un hôte, notamment en lui attribuant automatiquement une adresse IP et un masque de sous-réseau.

1) Sur *m1*, lancez le logiciel Wireshark : `wireshark &`

2) Sur *m2* et *m3*, modifiez avec l'éditeur de texte `nano`, le fichier `/etc/network/interfaces` en ajoutant les 2 lignes suivantes :
```
auto eth0
iface eth0 inet dhcp
```

3) Activez les interfaces réseau des *m2* et *m3* avec les commandes :
```
ifup eth0
```
Analysez les trames du protocole DHCP échangées entre le serveur et les clients. Quelle est l'adresse IP du serveur DHCP ? Que comprenez-vous du fonctionnement du protocole ? Notez les adresses IP attribuées à *m2* et *m3*.

4) Affichez le contenu des tables de routage de *m1*, *m2* et *m3* avec la commande :
```
route –n
```
Que constatez-vous ?

5) Désactivez l'interface de *m2* et changez son adresse MAC avec les commandes :
```
ifdown eth0
ifconfig eth0 hw ether 02:03:04:05:06:07
```
Activez à nouveau l'interface eth0. L'adresse IP attribuée est-elle la même que précédemment ?

Analyse de TCP à travers le protocole HTTP (*HyperText Transfert Protocol*)

- Assurez-vous que Wireshark est toujours lancé sur *m1*

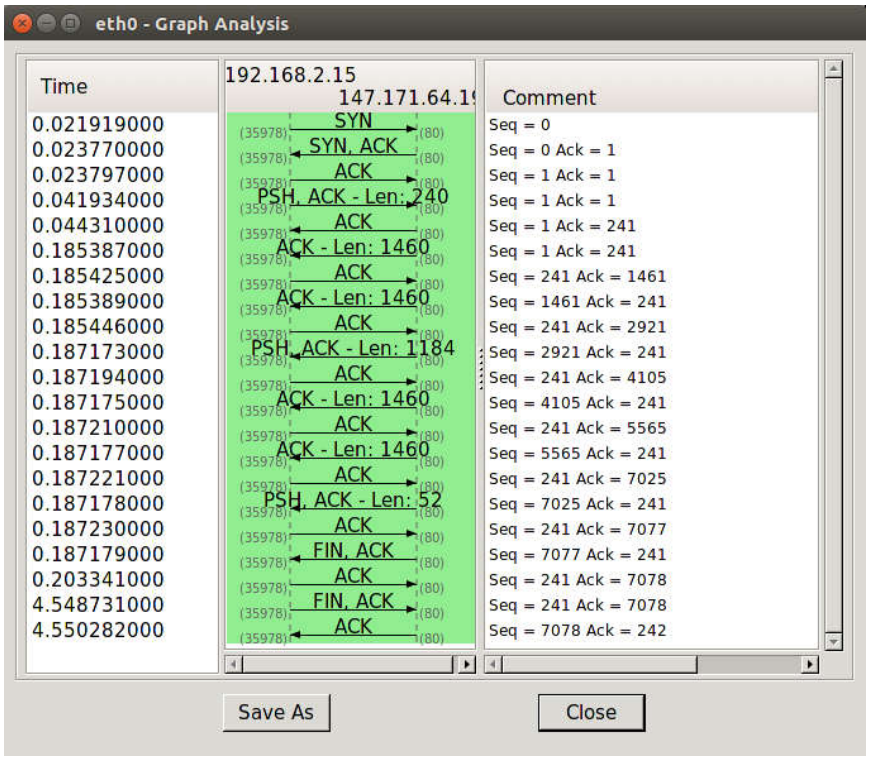
- Sur *m2*, lancez un navigateur en mode texte (`lynx`) et allez à l'URL :
  `lynx http://www.gnu.org`
  Quittez `lynx` avec le caractère « q », puis arrêtez la capture.

Suivi de session TCP

Observez les segments HTTP capturés et notamment les **poignées de main** de **début** (SYN, SYN-ACK, ACK) et de **fin de connexion** (FIN-ACK, FIN-ACK, ACK).

Lorsqu'un hôte initie une session TCP, son numéro de séquence initial est un nombre aléatoire compris entre 0 et 4 294 967 295. Cependant, les analyseurs de protocols comme Wireshark vont afficher des numéros de séquence et d'acquittement relatifs au flux de transmission au lieu des valeurs réelles[1].

Afin de mieux comprendre le fonctionnement des numéros de séquence et d'acquittement échangés lors d'une session TCP, il est possible d'utiliser la fonctionnalité *flow graphing* de Wireshark. Allez dans le menu « Statistics > Flow Graph... », sélectionnez « TCP flow » et cliquez sur OK. Wireshark va alors automatiquement construire le graphique suivant :



[1] L'affichage relatif des numéros de séquence peut être désactivé en allant dans le menu Edit > Preferences… > Protocols > TCP » et en décochant « Relative sequence numbers ».

Retransmission

Sur *m1*, simulez un serveur à l'écoute sur le port 80 à l'aide de la commande `netcat` (`man netcat`). Cet utilitaire Unix permet de transmettre (par défaut) et de recevoir (avec l'option *listen* `-l`) des données (lignes de texte) à travers une connexion réseau, en utilisant TCP (par défaut) ou UDP.

1) Lancez sur *m1* un processus `netcat` qui permettra de recevoir la requête d'un client HTTP lancé sur *m2* (sans cependant y répondre)
   ```
   netcat -l -p 80
   ```

2) Simulez un dysfonctionnement du réseau, en cliquant sur l'onglet « Anomalie » de Marionnet. Indiquez un taux de perte de 50 en entrée (inward) sur le port du hub *H1* sur lequel est branché *m1*.

3) Initiez une connexion HTTP depuis *m2* vers *m1*
   ```
   lynx http://192.168.2.254
   ```

   Vous devez apercevoir dans la fenêtre de *m1* la réception, de la part de `netcat`, de la requête HTTP. Bien entendu, `netcat` n'étant pas un serveur HTTP, il ne répondra pas à la requête laissant le client `lynx` en attente.

   Observez la retransmission des segments de données via le « Flow Graph ». Refaites de nouvelles captures, en modifiant le taux de perte.