Travaux Pratiques
Routage, SNAT, DNAT,
entre réseaux privés et réseaux publics

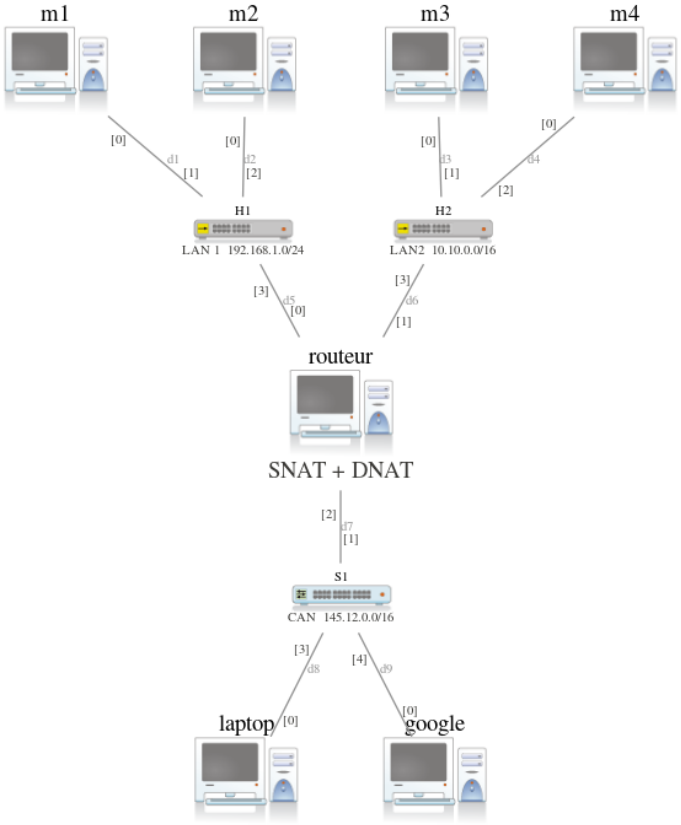
Séance de TP entièrement effectuée avec le logiciel Marionnet. Durée estimée : 1h30

Prérequis. Notions de routage, NAT (SNAT et DNAT) et leur interdépendance.

Câblage et configuration du réseau local

Deux machines, m_1 et m_2 et un concentrateur H_1 réalisent un réseau local $LAN_1 = \{m_1, m_2\}$ en 192.168.1.0/24. Deux autres machines m_3 et m_4 et un concentrateur H_2 réalisent un réseau local $LAN_2 = \{m_3, m_4\}$ en 10.10.0.0/16. Un troisième réseau CAN_1 (Campus Area Network) contiendra une machine appelée *laptop* et une autre appelée *google*. Une machine faisant office de routeur assurera la liaison (de niveau 3) entre LAN_1 (port 0), LAN_2 (port 1) et CAN_1 (port 2).

Distributions GNU/Linux. Utilisez la distribution Debian *wheezy* pour pouvoir lancer les commandes basiques de configuration et observation du réseau (`ifconfig`, `route`, `wireshark`, `tcpdump`, ...)



Attribution des IP. Par simplicité, la machine m_i aura l'adresse 192.168.1.i ou 10.10.0.i selon le réseau d'appartenance. Le routeur *routeur* prend la dernière adresse disponible sur les deux réseaux privés. Concernant le réseau CAN (145.12.0.0/16), la machine *laptop* prendra le 145.12.0.42, *google* prendra le 145.12.0.84, et le *routeur* (qui est une machine ordinaire sur le réseau public) prendra le 145.12.0.53 sur le port 2 (`eth2`).

Première partie
Routage

Configurer la machine *routeur* et définissez-la comme passerelle pour toutes les autres machines du réseaux, même si cela est **abusif** (et non souhaité) pour le réseau public (pour lequel *routeur* n'est pas une passerelle mais une machine tout à fait ordinaire du réseau CAN). Testez avec la commande `ping` que toutes les machines puissent communiquer avec toutes les autres.

Deuxième partie
SNAT

Activez des services réseau : par exemple les services TCP `apache2` sur *google* et `ssh` sur *laptop*. Configurez le SNAT sur *routeur* avec la commande `iptables` de façon que tout paquet provenant des réseaux privés ait comme source (apparente) *routeur* aux yeux de *laptop* et *google*. Capturez les trames avec `wireshark` sur ces deux machines pour vérifier que le SNAT fonctionne correctement.

Exemple à adapter à notre réseau :

```
iptables -t nat -A POSTROUTING -o eth3 -s 172.23.5.0/24 -j MASQUERADE
```

Autres commandes utiles :

— pour afficher la liste (`-L`) des règles de NAT actuellement définies :

```
iptables -t nat -L              # ajouter -v pour un rendu plus détaillé (verbeux)
```

— pour effacer une règle définie précédemment : on reprend la même commande en remplaçant `-A` (append) par `-D` (delete) ; pour l'exemple ci-dessus, cela donnerait :

```
iptables -t nat -D POSTROUTING -o eth3 -s 172.23.5.0/24 -j MASQUERADE
```

Troisième partie
DNAT

Activez des services réseau dans les réseaux locaux : par exemple les services `apache2` (`HTTP/TCP[80]`) sur $m1$ et `ssh` (`SSH/TCP[22]`) sur $m4$. Configurez le DNAT sur *routeur* de façon que tout paquet provenant du réseau public CAN ait comme destinataire (apparent) *routeur* (aux yeux de *laptop* et *google*). Capturez les trames avec `wireshark` sur les réseaux privés pour vérifier que le DNAT fonctionne correctement. Enlevez donc la route par défaut *routeur* (qui était abusives) des machines *laptop* et *google*. Tout devrait continuer de fonctionner (vérifiez à nouveau avec `wireshark`).

Exemples à adapter à notre réseau :

```
iptables -t nat -A PREROUTING -i eth3 -p udp --dport 53   -j DNAT --to 172.23.5.98
iptables -t nat -A PREROUTING -i eth3 -p udp --dport 5353 -j DNAT --to 172.23.5.99:53
```

Une fois les services web (HTTP) et de connexion à distance (SSH) fonctionnant et correctement accessibles (DNAT+SNAT) depuis le CAN, activez un deuxième service `apache2` (HTTP/TCP[80]) sur $m3$ et donnez accès aux machines du CAN, c'est-à-dire *laptop* et *google*, à ce deuxième site web accessible par le port (fictif) `8080` du *routeur* (pour distinguer les sites web, modifiez les pages d'accueil, c'est-à-dire les fichiers `/var/www/index.html`, sur $m1$ et $m3$).