

## Pendant ce TD,

- vous apprendrez à configurer des machines en réseau,
- vous utiliserez les commandes *ping* et *tcpdump* pour observer les communications,
- vous diagnostiquerez quelques problèmes courants,
- vous aborderez quelques problématiques de routage : déport d'un sous-réseau, fédération de réseaux.

Le TP sera réalisé à l'aide de machines virtuelles Linux (distribution Slackware).

**Copiez chez vous** le script `tp3-reseau.sh` qui se trouve dans le répertoire

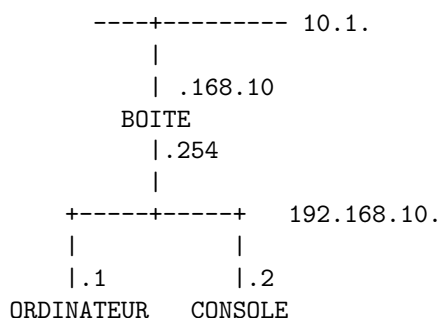
`~/Bibliotheque/S3-ASR3-Réseaux/TD3_Routage_UML`

où vous trouverez d'autres informations utiles.

**Prenez scrupuleusement des notes** pendant la séance, et envoyez-les à votre enseignant en fin de TD.

## 1 Un réseau simple de trois machines

On simule ici le parc informatique d'un particulier, avec 3 appareils : une "boîte" (routeur) prêtée ou vendue par son fournisseur d'accès, un ordinateur pour travailler et l'autre pour jouer.



Le fournisseur a accordé à ce client l'usage de la plage d'adresses 192.168.10.0 à 192.168.10.255, sous-réseau dont le masque est 255.255.255.0. Le dernier octet de l'adresse IP figure sur le schéma (exemple 192.168.10.2 pour la console).

Les boîtes sont reliées entre-elles par le réseau 10.0.0.0/8.

1. ce sont les noms des fichiers contenant les images des disques des machines virtuelles. À ne pas confondre avec le nom que l'utilisateur choisit de donner à sa machine (darkstar par défaut pour la distribution Slackware. Le préfixe tp3 vous permettra de faire le ménage dans vos fichiers plus tard

## 2 Configuration "à la main"

**1** Lancez le script `reseau-tp3.sh -base` qui fera apparaître 3 machines virtuelles, dans des fenêtres appelées `tp3boite`, `tp3ordinateur`, `tp3console`.<sup>1</sup>

**2** Avec la commande *ifconfig*, attribuez les adresses IP aux machines.

La configuration voulue est la suivante

```

BOITE      interface eth0
           adresse IP 192.168.10.254
           masque 255.255.255.0
           interface eth1
           adresse IP 10.1.168.10
           masque 255.0.0.0
           routes
           par défaut: à voir plus tard

ORDINATEUR interface eth0
           adresse IP 192.168.10.1
           masque 255.255.255.0
           routes
           par défaut: par boite

CONSOLE    interface eth0
           adresse IP 192.168.10.2
           masque 255.255.255.0
           routes
           par défaut: par boite
  
```

**3** Avec la commande *ping -c3 destination* qui envoie 3 paquets *ping*, vérifiez que chaque machine est bien en état de contacter les deux autres.

**4** Lancez l'utilitaire *tcpdump* sur *boite*, pour observer soigneusement le trafic quand, depuis l'ordinateur, on fait un *ping* vers

1. la boite
2. la console
3. 192.168.10.123 (adresse appartenant au sous-réseau, qui ne répond pas)
4. 10.1.99.99 (dans le réseau du FAI)

#### 5. 1.2.3.4 (hors du réseau)

Notez le dialogue qui se produit (arp + ping + ...), avec les adresses ARP et IP impliquées. **Attention** : avant chaque expérimentation, prenez bien soin de vider le cache ARP (commande `arp -d adresse`) d'*ordinateur*.

**5** Faites `halt` sur chaque machine virtuelle. Quand elles sont toutes arrêtées (attendez patiemment), redémarrez le script du réseau, faites *ping* d'une machine à une autre. Que constatez-vous ?

## 3 Configuration permanente

Au démarrage, une machine Unix lance un grand nombre de scripts<sup>2</sup> qui démarrent divers services. Dans la distribution Slackware, le démarrage du réseau est confié à `/etc/rc.d/rc.inet1` qui lance *ifconfig*, *route* etc. sous le contrôle du fichier de paramètres `/etc/rc.d/rc.inet1.conf`. Extrait :

```
# Config information for eth0:
IPADDR[0]=" "
NETMASK[0]=" "
USE_DHCP[0]=" "
DHCP_HOSTNAME[0]=" "
```

```
# Config information for eth1:
IPADDR[1]=" "
...
# Default gateway IP address:
GATEWAY=" "
```

Remplir ce fichier à la main ne pose guère de difficultés, mais ce n'est pas très confortable, d'autant que d'autres fichiers doivent être renseignés (`/etc/HOSTNAME`, `/etc/hosts`) par la même occasion.

**6** Que contiennent ces deux fichiers ?

### 3.1 Administration : netconfig

**7** Avec le script d'administration *netconfig*, configurez ces trois machines, avec les noms *boite*, *ordinateur* et *console* dans le domaine *maison*, sans DNS. Avant de lancer *netconfig*, définissez le bon type de terminal (*xterm*) :

```
root@darkstar: TERM=xterm
root@darkstar: netconfig
...
```

**8** La configuration est prise en compte au redémarrage, ou quand on le demande explicitement<sup>3</sup> par

```
/etc/rc.d/rc.inet1 restart
```

Après avoir redémarré les machines vérifiez que

- 2. des shell-scripts, comme on les apprend en première année
- 3. mais pas le `hostname`

1. les noms des machines s'affichent dans la bannière de login (ils sont pris dans le fichier *HOSTNAME*)
2. les machines répondent au *ping* (dans les 6 directions).

## 3.2 Résolution des noms

**9** Depuis *ordinateur* faites *ping* vers

- 127.0.0.1, localhost
- 192.168.10.1, *ordinateur*, *ordinateur.maison*
- 192.168.10.2, *console*, *console.maison*

vous constaterez que les noms des autres machines ne sont pas connus par *ordinateur*.

C'est en principe le rôle d'un DNS (*domain name server* = serveur de noms) que de fournir, à la demande, le numéro IP qui correspond à un nom. Il n'y a pas de DNS sur notre réseau, nous allons compléter le fichier `/etc/hosts` de chaque machine.

**10** Sur *ordinateur*, ajoutez ces deux lignes :

```
192.168.10.254  boite  boite.maison
192.168.10.2    console console.maison
```

et vérifiez qu'on peut maintenant faire des *ping* vers *boite* et *console*. Faire la même chose sur les 2 autres machines.

## 3.3 Utilisation des services

**11** Sur *ordinateur* et *console*, enregistrez un utilisateur :

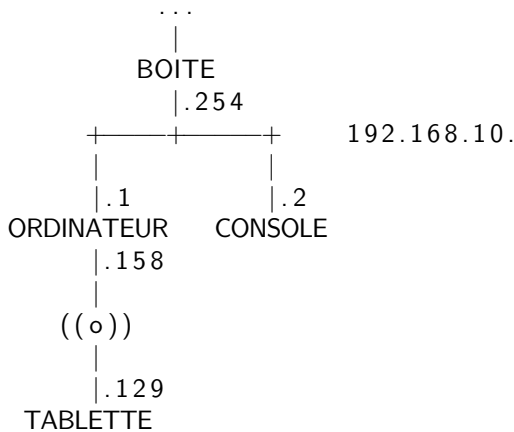
```
useradd -m marcel ; passwd marcel
```

et vérifiez

- qu'il peut se connecter sur une machine
- et qu'il a accès à l'autre par `ssh marcel@nom-de-la-machine`

## 4 Ajout d'un sous-réseau

Pour lire son journal le matin, le joyeux abonné achète une tablette qu'il connecte à son PC par un adaptateur Wifi USB, qui se comporte comme une carte réseau supplémentaire (eth1), connectée à un réseau "aérien".



Il décide d'affecter à ce réseau Wifi les adresses 192.168.10.128 à 192.168.10.159, c'est-à-dire le sous-réseau 192.168.10.128/27 de masque 255.255.255.224.<sup>4</sup>

### 4.1 Configuration

**12** Complétez

```
ORDINATEUR interface eth0
             adresse IP 192.168.10.1
             masque 255.255.255.0
             interface eth1 (wifi)
             adresse
             masque
             routes
             par défaut: par boite

TABLETTE interface eth0
           adresse
           masque
           routes
           par défaut:
```

**13** Pour que *ordinateur* fasse passerelle entre deux réseaux, il faut rendre exécutable un script qui active cette fonctionnalité du noyau. Dans `/etc/rc.d`, faites

```
chmod +x rc.ip_forward
```

sur *ordinateur* et *boite* pour la même raison.

**14** Faites `halt` sur toutes les machines, puis lancez le script `reseau-tp3 -wifi`

**15** Configurez *tablette* et modifiez la configuration de *ordinateur* pour que ces deux machines puissent communiquer.

### 4.2 Un petit problème

**16** Remarquez que, depuis *tablette*, un *ping* vers *console* "ne répond pas". En faisant tourner *tcpdump* sur *console* (et éventuellement sur l'interface `eth1` de l'*ordinateur*), déterminez exactement le scénario de transmission des paquets ARP et PING.

**17** Montrez qu'on peut remédier à ce problème en ajoutant une route<sup>5</sup> sur *console*. Remarquez que le même problème se pose pour *boite*, et toutes les machines de ce sous-réseau.

### 4.3 Solutions alternatives

A lire plus tard, si il vous reste du temps.

Dans la situation ci-dessus une partie des adresses allouées a été affecté à un sous-réseau différent, ce qui les rend inaccessible par le routage "normal" sur un réseau local. Nous avons vu qu'un ajout de routes apportait une solution, mais il y a un inconvénient : il faut configurer en conséquence toutes les machines du réseau local.

Il existe des alternatives :

- **routage dynamique** : faire en sorte que les postes qui se connectent au réseau se configurent en fonction d'une liste de routes publiées par un serveur. C'est une des fonctions que peut assurer un serveur DHCP.<sup>6</sup>
- **proxy ARP** : mandater la machine *ordinateur* pour qu'elle réponde aux requêtes ARP à la place de *tablette*, en donnant sa propre adresse MAC. Ainsi, quand *console* envoie un paquet (ping, ip, ...) à *tablette*
  - l'adresse destinataire MAC de la trame mise sur le réseau local par *console* est celle de la carte *eth0* de *ordinateur*,
  - *ordinateur* se saisit donc de la trame, analyse son contenu et, constatant que l'adresse IP destinataire est celle de *tablette*, route le paquet sur le sous-réseau wifi.

Le "déport" du sous-réseau est alors complètement transparent. Exemple de commande à lancer sur *ordinateur* :

```
# requêtes ARP : répondre à la place
# de tablette sur eth0
arp -Ds 192.168.10.129 eth0 pub
```

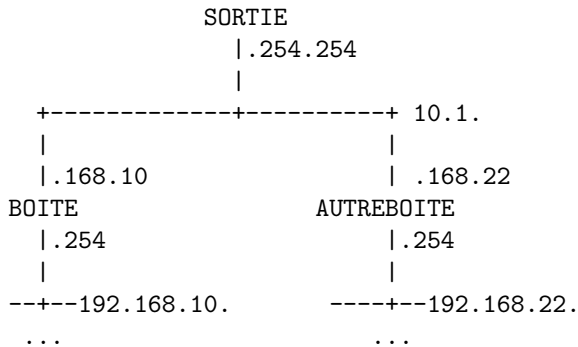
4. ce qui autorisera l'utilisation d'une trentaine d'appareils WIFI simultanément lors des fêtes de famille.

5. le fichier `/etc/rc.d/rc.local` est approprié

6. les "boîtes" de connexion à Internet contiennent un serveur DHCP, qui distribue les adresses IP aux machines qui se connectent

## 5 Intégration dans le réseau FAI

Sur le réseau du FAI, les numéros de chaque boîte est formé à partir du numéro de réseau abonné : 10.1.168.10 pour la boîte considérée, 10.1.168.22 pour le réseau 192.168.22.0/24 etc. Un routeur 10.254.254.254 conduit vers le reste du réseau mondial.



**18** Faites `halt` sur chaque machine virtuelle puis lancez la commande `reseau-tp3.sh -fai`

**19** Configurez la communication entre boîte et sortie

```
BOITE
interface eth0 inchangée
interface eth1
    adresse 10.1.168.10
    masque 255.0.0.0
routes
    par défaut : vers 10.254.254.254

SORTIE
interface eth0
    adresse 10.254.254.254
    masque 255.0.0.0
routes
```

par défaut : non  
vers 192.168.10.0/24, via 10.1.168.10  
vers ...

**20** Constatez que la communication passe de *sortie* vers *ordinateur* et *tablette*, et inversement.

**21** Faites la même chose pour *autreboite*, qui fait passer entre le réseau fournisseur et un autre réseau d'abonné (adresses IP 192.168.22.254/24 et 10.1.168.22/8)

**22** Constatez que depuis *tablette* on fait ping avec succès sur les adresses 192.168.22.254 et 10.1.168.22, alors qu'il n'y a pas de route explicite d'un abonné à l'autre.

**23** Avec `tcpdump` sur *sortie*, observez les paquets qui passent lors d'un ping d'un abonné à l'autre.

**24** Ce "rebond" sur le routeur peut vous paraître inefficace. Regardez le contenu des tables de routage des boîtes (`route -C`). Que voyez-vous ?

**En réalité**, les fournisseurs d'accès (pour particuliers) offrent seulement une seule adresse IP par abonné, ce qui simplifie la gestion<sup>7</sup>.

Dans les premiers temps d'internet, les contrats d'abonnement interdisaient d'ailleurs de raccorder des appareils à celui qui était officiellement connecté...

Pour partager la ligne, il faut que l'appareil connecté serve d'intermédiaire, de relais, pour les communications venant des autres appareils. Ceci peut se faire à différent niveaux :

- applicatif : on parle alors de *mandataire* (proxy) ;
- réseau : le relais transmet les communications vers l'extérieur en faisant croire qu'il en est l'origine, et fait suivre les réponses. C'est le *masquering*, ou *NAT* (*network address translation*), qui sera vu au S4.

7. et économise les adresses IP