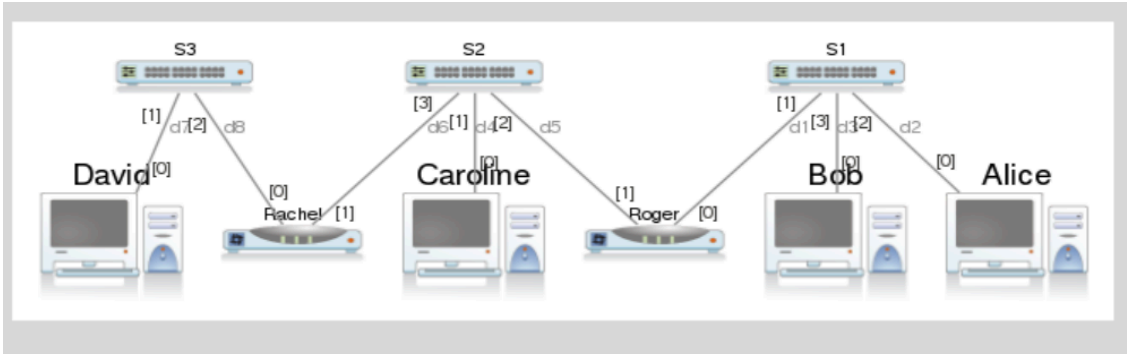


- un réseau d'adresse 192.168.1.0/24 comportant Alice, Bob et Roger.
- un réseau d'adresse 172.16.0.0/16 comportant Roger, Caroline et Rachel.
- un réseau d'adresse 10.0.0.0/8 comportant Rachel et David.



2. Premiers réseaux (3 pts)

Créez votre projet Marionnet et ajoutez-y les machines Alice, Bob, Roger et Caroline. Configurez machines de façon à ce qu'elles puissent toutes communiquer entre elles

Alice	Bob	Caroline
<pre> ifconfig eth0 up ifconfig eth0 192.168.1.1/24 Alice zebra Roger : telnet 192.168.1.254 2601 enable configure terminal interface eth1 ip addr 172.16.0.254/16 end copy running-config startup-config (sauvegarde) exit Routage route add default gw 192.168.1.254 dev eth0 </pre>	<pre> //comme Alice ifconfig eth0 up ifconfig eth0 192.168.1.2/24 Routage comme Alice </pre>	<pre> ifconfig eth0 up ifconfig eth0 172.16.0.1/16 route add default gw 172.16.0.254 dev eth0 // route add -net res.des gw passerelle dev interface </pre>

3. DHCP (3 pts)

Configurez et démarrez sur Alice un service DHCP qui lui permet d'attribuer dynamiquement une adresse IPv4 à Bob. Assurez-vous du bon fonctionnement du service.

<pre> Alice : mv /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.old + touch /etc/dhcp/dhcpd.conf + vi /etc/dhcp/dhcpd.conf ddns-update-style none; default-lease-time 600; max-lease-time 7200; authoritative; subnet 192.168.1.0 netmask 255.255.255.0 { range 192.168.1.3 192.168.1.10; } TEST : dhcpd (si ça affiche : Sending on Socket/fallback/fallback-net, c'est que ça marche) Bob : dhclient eth0 (ça doit afficher un message comme ça : bound to 192.168.1.1 – renewal in 300 seconds) </pre>
--

4. DNS (3 pts)

Configurez et démarrez sur Caroline un service DNS (zone simple, pas de résolution inverse) qui définit une zone tp-reseau. Attribuez à Alice le nom dhcp-serv, à Bob le nom client, à Caroline le nom dns-serv et à Roger le nom routeur. Assurez-vous qu'il soit consultable depuis toutes les machines en place.

<p>Caroline :</p> <pre> vi /etc/bind/named.conf.local Dans vi : zone "tp-reseau." { type master; </pre>

5.

```
file "/etc/bind/db.tp-reseau";
};
```

named-checkconf : vérification du fichier de configuration -> pas de message = ça marche

vi /etc/bind/db.tp-reseau

Dans vi :

```
@ IN SOA serv-dns.tp-reseau. root.serv-dns.tp-reseau. (
    2023041301 ; serial | 28800 ; refresh | 14400 ; retry | 36000 ; expire | 86400 ; default_ttl
)
@ IN NS serv-dns.tp-reseau.
serv-dns IN A 172.16.0.1 (ipv4_Caroline)
client IN A 192.168.1.2 (ipv4_Bob)
dhcp-client IN A 192.168.1.1 (ipv4_Alice)
routeur IN A 192.168.1.254 (ipv4_Roger_eth0)
routeur IN A 172.16.0.254 (ipv4_Roger_eth1)
```

Caroline : vérification du fichier, pas de message erreur et ça affiche "ok" = ça marche PUIS

named-checkzone tp-reseau /etc/bind/db.tp-reseau ET Lancement du dns avec -> named

Configuration Client : Alice et Bob

vi /etc/resolv.conf

```
nameserver ipv4_Caroline
search tp-reseau
```

Alice et Roger tester la commande suivante : pas de msg d'erreur = 👍

host serv-dns.tp-reseau /host client /host dhcp-client

Routage Avancé (3 pts)

Ajoutez les machines Rachel et David, créant le nouveau réseau 10.0.0.0/8. Faites les configurations nécessaires pour que David puisse communiquer avec Alice et consulter le DNS de Caroline.

Mettre en place un deuxième ordinateur -> David Et un deuxième routeur -> Rachel (On lie Rachel au switch 2 avec des câbles droits).(Entre David et Rachel, soit un câble croisé ou un Switch 3 (à ajouter) puis deux câbles droits).

Quand on créer le routeur Rachel on change l'adresse IP par défaut par celle du réseau 10.0.0.0/8 en mettant l'adresse IP : 10.0.0.254/8. (En gros faut appuyer sur le bouton "modifier" du routeur Rachel sur l'interface de Marionnet puis saisir cet IP sur le port 0)

On peut également faire avec l'adresse 172.16.0.253/16, ça dépend de la façon dont vous voulez faire le routage.

David :

```
ifconfig eth0 up
ifconfig eth0 10.0.0.1 /8
telnet 10.0.0.254 2601
enable
configure terminal
interface eth1
ip addr 172.16.0.253 (par exemple)
end
exit
puis faire un :
route add default gw passerelle dev interface
```

soit :

David = route add default gw 10.0.0.254 dev eth0

Caroline = route add default gw 172.16.0.253 dev eth0

Puis on revient sur Zebra et on ajoute les routes sur Rachel et Roger.

Depuis Caroline :

```
telnet 172.16.0.253 2601
mdp : zebra
enable
mdp : zebra
configure terminal
ip route 192.168.1.0/24 172.16.0.254
end
exit
```

Depuis Alice ou Bob :

```
telnet 192.168.1.254 2601
mdp : zebra
enable
mdp : zebra
configure terminal
ip route 10.0.0.0/8 172.16.0.253
end
exit
```

Tester avec des pings et vérifiez les câbles + port de vos routeurs. Pour pouvoir consulter le DNS de Caroline avec David c'est simple :

```
vi /etc/resolv.conf
nameserver 172.16.0.1
```

search tp-reseau

puis faire des tests avec la commande host comme sur l'étape du DNS. (David et Caroline) + test host

6. Pare-feu (2 pts)

Caroline :

Règle(s) :

```
# Politique par défaut : refuser tout le trafic entrant
iptables -P INPUT DROP

# Autoriser les réponses DNS (UDP)
iptables -A INPUT -p udp --sport 53 -j ACCEPT

# Autoriser les requêtes DNS (UDP)
iptables -A INPUT -p udp --dport 53 -j ACCEPT

# Autoriser les requêtes DNS (TCP)
iptables -A INPUT -p tcp --dport 53 -j ACCEPT

# Rejeter tout autre trafic entrant
iptables -A INPUT -j DROP
```

Partie révision

COURS 1 :

Protocole : Normes de communication entre différents équipements d'un réseau.

Pile de protocole : Organisation des protocoles suivant un « empilement » : chaque protocole collabore avec un protocole de niveau inférieur et supérieur.

Open systems interconnection (OSI) : interconnexion de systèmes ouverts.

- Système ISO 7498
- Norme théorique d'architecture réseau reposant sur une pile de protocole

Ip (internet protocol) : permet d'assigner à chaque machine une adresse présumée unique comprenant un identifiant de réseau et un identifiant d'interface.

Taille IPv4 : 32 bits

Taille IPv6 : 64 bits

COURS 2 :

Dynamic Host Configuration Protocol (DHCP) : Protocole de configuration dynamique d'hôte.

- Attribut automatique d'une adresse IP sur un réseau
- Basé sur un modèle client-serveur

COURS 3 :

Pare-feu : Logiciel de sécurité permettant d'autoriser ou d'interdire certaines communications dans un ou à travers un point de réseau donné.

Les types de pare-feu :

Pare-feu sans états :

- Plus ancien et plus rudimentaire que les autres versions
- Fonctionne en suivant une liste de règles
- Ne permet pas une grande finesse de configuration

Pare-feu à états :

- Tire parti de la notion de connexion des protocoles basés sur TCP
- Vérifie pour chaque message sa conformité avec l'échange en cours
- Permet aussi une configuration plus rapide pour les échanges UDP

Pare-feu applicatifs :

- Vérifie la conformité des échanges par rapport au protocole prévu
- Consomme plus de ressources
- Mais nécessaire pour pallier aux contournement pare-feu par tunnel TCP et pour la gestion de certains protocoles (FTP)

Pare-feu à identifiant :

- Repose sur une identification de l'utilisateur
- Permet de définir des droits par utilisateur et non plus par machine

Quelques informations supplémentaires sur le pare-feu :

- Il ne s'applique qu'à l'endroit où il est installé
- Ne protège que la machine hôte, pas le reste du réseau

COURS 4 :

Transmission Control Protocol (TCP) : Protocole de contrôle de transmission.

- Permet l'échange fiable entre deux équipements d'un réseau
- Se situe sur la couche transport du modèle OSI
- Manipule des segments d'information

Session TCP : 3 étapes fondamentales

- Établir la connexion
- Transférer les données
- Clore la connexion proprement

User Datagram Protocol (UDP) : protocole de datagramme utilisateur.

- Permet l'échange rapide d'information entre un ou plusieurs équipements d'un réseau
 - Se situe sur la couche transport du modèle OSI
 - Manipule des datagrammes d'information
-

Truc pas mis :

- table de routage
- définition d'un numéro de port