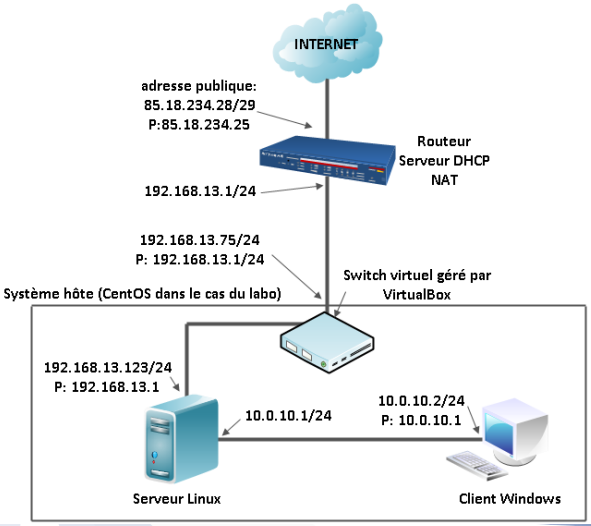
Partage de connexion Internet (rédigé pour Ubuntu Server)

Hainaut Patrick 2016

But de cette présentation

- Maintenant que nous avons configuré le rôle serveur DHCP sur notre serveur Linux, donnons l'accès à Internet aux clients du réseau local
- Cela permettra de voir une première règle de firewall

Schéma de principe



Le serveur Linux est client DHCP sur l'interface réseau reliée au routeur (enp0s3 par exemple) et serveur DHCP sur l'autre (enp0s8 par exemple)

Le PC client reçoit ses paramètres IP du serveur Linux.

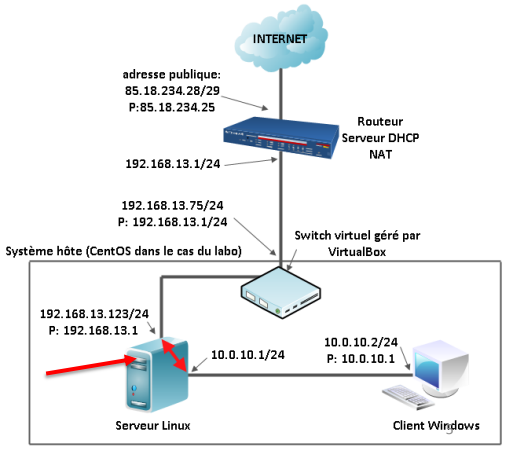
Partage de connexion Internet

- On voudrait pouvoir à partir du PC client, surfer sur le net …
- Pour cela, il va nous falloir configurer deux choses sur notre serveur Linux
  - Activer le passage des trames IP d'une interface réseau à l'autre
  - Faire du NAT (Network Address Translation)

Passage des trames IP d'une interface à l'autre

- Linux possède un interrupteur général autorisant ou non le passage d'une trame IP d'une carte réseau à l'autre (entre enp0s3 et enp0s8)

- Si l'interrupteur est à 'off ', les demandes en provenance du PC client, arrivant dans notre exemple sur enp0s8, ne seront jamais transmises à enp0s3, reliée au réseau extérieur



Passage des trames IP d'une interface à l'autre

- Cet interrupteur est contrôlé via la variable **ip_forward**

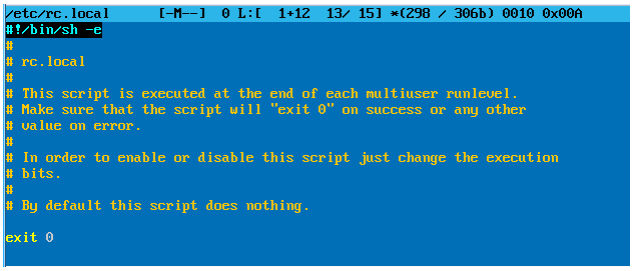
- cat /proc/sys/net/ipv4/ip_forward permet de voir l'état de l'interrupteur:
  0: pas de passage
  1: passage autorisé selon la configuration

- echo 1 > /proc/sys/net/ipv4/ip_forward permet d'activer cet interrupteur

- Pour que ce réglage perdure, il faut copier la dernière commande dans **/etc/rc.local**

Passage des trames IP d'une interface à l'autre

- Il faut donc éditer le fichier **/etc/rc.local**



- Attention, la commande doit être tapée avant le exit 0 (qui signifie la sortie du fichier)

Principe du NAT dynamique

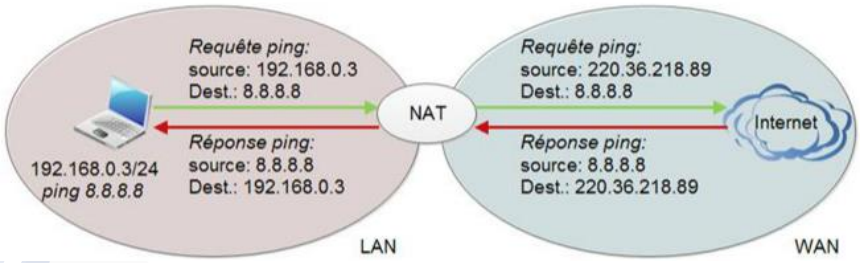
- Pour surfer sur Internet, il faut une adresse IP publique
- Dans un réseau local, en général, une seule machine est reliée à Internet, les autres PC passant par elle (on l'appelle d'ailleurs 'passerelle' ou 'gateway' en anglais) pour sortir
- Cette passerelle sera généralement constituée par un routeur ou un serveur assumant la fonction de routage
- Un PC du réseau local passant par cette passerelle ne pourra pas accéder directement à Internet car son adresse IP privée ne lui permet pas

Principe du NAT dynamique

- Il faut donc un mécanisme qui va 'échanger' l'adresse IP privée du PC client par une adresse IP publique (celle de la passerelle)
- C'est le mécanisme de translation d'adresse (NAT en anglais)
- Quand la passerelle revient avec les données d'Internet, elle échange de nouveau les adresses pour transmettre ces données au PC client
- La passerelle contient une table dynamique qui lui permet de savoir qui est à l'origine de quelle requête et donc remettre la réponse au bon destinataire sur le réseau local

Principe du NAT dynamique

- Tout le réseau local peut donc "surfer" sur Internet avec une seule machine effectivement connectée à Internet (la passerelle)

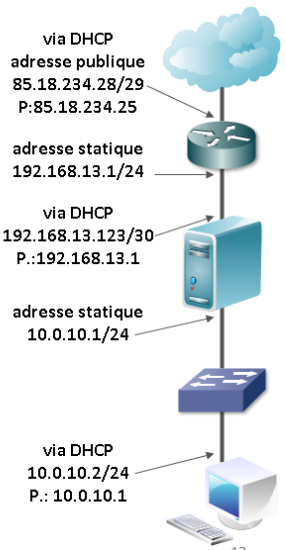


Principe du NAT dynamique

- En plus de faire de la translation d'adresses, le NAT dynamique utilise le mécanisme de translation de port (**PAT** - *Port Address Translation*)
- Cela affecte un port source différent à chaque requête de telle manière à pouvoir maintenir une correspondance entre les requêtes provenant du réseau interne et les réponses des machines sur Internet, toutes adressées à l'adresse IP du routeur.

Principe du NAT dynamique

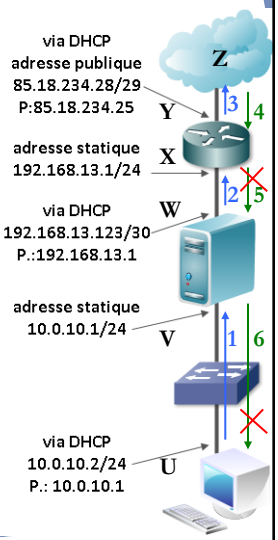
- Dans notre cas, l'adresse IP du serveur Linux du coté WAN est une adresse privée (dans notre exemple: 192.168.7.45)
- Théoriquement, on ne doit donc pas mettre en œuvre le NAT sur ce serveur
- Le NAT sera mis en œuvre sur le routeur
- Pourtant, si vous faites un ping d'une adresse publique (8.8.8.8), ça ne fonctionne pas …



Principe du NAT dynamique

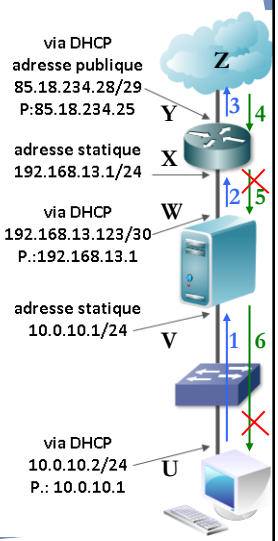
- Examinons les trames Ethernet échangées …

	Trame Ethernet		Paquet IP		
1	MAC des: V	MAC src: U	IP des:8.8.8.8	IP src:10.0.10.2	Datas
2	MAC des: X	MAC src: W	IP des:8.8.8.8	IP src:10.0.10.2	Datas
3	MAC des: Z	MAC src: Y	IP des:8.8.8.8	IP src:85.20.39.163	Datas
4	MAC des: Y	MAC src: Z	IP des:85.20.39.163	IP src:8.8.8.8	Datas
5	MAC des: ?	MAC src: X	IP des:10.0.10.2	IP src:8.8.8.8	Datas
6	MAC des:	MAC src:	IP des:	IP src:	Datas



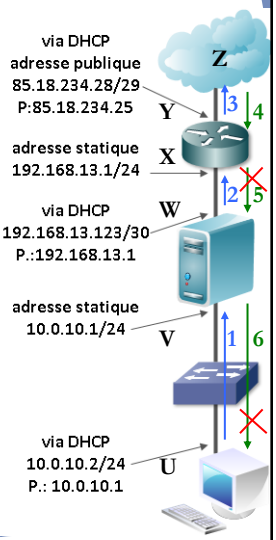
Principe du NAT dynamique

- Pour l'aller, pas de problème
- Mais pour le retour, le routeur qui applique le NAT dans l'autre sens, et qui construit donc une trame Ethernet avec 10.0.10.2 comme IP de destination, n'a pas de route dans sa table de routage pour le réseau 10.0.10.0/24 …
- Il a une route par défaut vers Internet et une autre vers le réseau 192.168.7.0/24
- Il laisse donc tomber le paquet IP



Principe du NAT dynamique

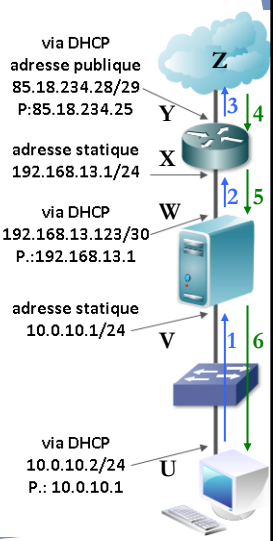
- Pour que ça puisse fonctionner, deux méthodes:
  - Soit on rajoute une route dans le routeur vers le réseau 10.0.10.0/24 ou vers le réseau 10.0.0.0/8 pour tenir compte de tous les sous-réseaux éventuels
  - Soit on rajoute une règle de NAT sur le serveur Linux
  - La première solution est plus professionnelle (car le NAT est prévu pour échanger une IP privé par une IP publique, ce qui ne sera pas le cas ici) mais on doit avoir accès au routeur …



Principe du NAT dynamique

- Si on active le NAT sur le serveur Linux …

	Trame Ethernet		Paquet IP		
1	MAC des: V	MAC src: U	IP des:8.8.8.8	IP src:10.0.10.2	Datas
2	MAC des: X	MAC src: W	IP des:8.8.8.8	IP src:192.168.7.45	Datas
3	MAC des: Z	MAC src: Y	IP des:8.8.8.8	IP src:85.20.39.163	Datas
4	MAC des: Y	MAC src: Z	IP des:85.20.39.163	IP src:8.8.8.8	Datas
5	MAC des: W	MAC src: X	IP des:192.168.7.45	IP src:8.8.8.8	Datas
6	MAC des: U	MAC src: V	IP des:10.0.10.2	IP src:8.8.8.8	Datas



Mise en œuvre du Nat

- C'est notre serveur Linux qui jouera le rôle de passerelle
- On active le NAT (ou masquerading) par une règle de firewall:
  iptables –t nat –A POSTROUTING –o enp0s3 –j MASQUERADE
- enp0s3 est l'interface reliée à Internet (adaptez suivant votre configuration)
- Pour que ce réglage perdure, il faut copier la dernière commande dans **/etc/rc.local**

Exécution du rc.local

- Le /etc/rc.local devrait ressembler à ceci:

```
/etc/rc.local       [-M--] 52 L:[  1+13   14/ 16] *(389 / 397b) 0010 0x00A
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
exit 0
```

- Pour l'éxécuter, tapez simplement /etc/rc.local
- Au redémarrage du serveur, il sera exécuté automatiquement

Si on veut être plus professionnel …

- La méthode avec **rc.local** ne fonctionne pas avec toutes les distributions car ce fichier n'est pas forcément lu lors d'un redémarrage du réseau
- Pour changer la variable **ip_forward**, vous pouvez éditer le fichier **/etc/sysctl.conf** et décommenter la ligne **#net.ipv4.ip_forward=1** en enlevant le dièse
- Pour recharger la config.: sysctl -p

Si on veut être plus professionnel …

- Pour la règle de NAT (et toutes les autres règles iptables), on peut taper la commande directement dans la CLI

  iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE

  et sauver cette commande ou cet ensemble de commandes par iptables-save > */etc/iptables-rules*

  où */etc/iptables-rules* peut être n'importe quel nom de fichier (et emplacement)

Si on veut être plus professionnel …

- Il reste à indiquer la commande

  pre-up iptables-restore < */etc/iptables-rules*

  à la fin du fichier **/etc/network/interfaces**

- Au redémarrage, les réglages concernant ip_forward et le nat perdurent

Test sur le PC client

- Dans une invite de commande, un ping vers une adresse externe comme 8.8.8.8 (DNS de Google) devrait passer

- Un ping vers www.google.be devrait fonctionner aussi pour peu qu'un serveur DNS correct soit spécifié dans la configuration de notre serveur DHCP

Serveur DNS

- C'est la directive
  option domain-name-servers <IP du serveur DNS>;

  du fichier **/etc/dhcp/dhcpd.conf** qui permet de spécifier l'adresse du serveur DNS

  Ex.: option domain-name-servers 8.8.8.8;

  Cela oblige à connaître l'IP d'un serveur DNS externe …

Installer son propre serveur DNS

- Si vous voulez que les requêtes DNS du client passe par votre serveur Linux, vous pouvez installer le paquet **bind9**
- L'adresse du serveur DNS à spécifier dans la config du serveur DHCP (/etc/dhcp/dhcpd.conf) est maintenant celle du serveur Linux
- Dans notre exemple, cela donne:
  option domain-name-servers 10.0.10.1;

Configuration de Bind9

- Pour ce qu'on veut faire actuellement (partage de connexion Internet), la configuration par défaut de **bind9** se suffit à elle-même
- En effet, les adresses des serveurs DNS racines sont présentes par défaut , donc notre serveur DNS interne fera bien le lien avec tout site internet

Test sur le PC client

- Après avoir renouvelé les paramètres IP (**ipconfig /release** suivi de **ipconfig /renew**), testez l'accès au réseau extérieur (**ping 8.8.8.8**) et la résolution de noms (**ping www.google.be**)
- Ouvrez votre navigateur Internet favori et surfez … (n'oubliez pas de renseigner le proxy éventuel …)

Test sur le PC client

- A noter que ces manipulations peuvent fonctionner aussi avec un PC physique ou un réseau physique relié à une carte réseau réelle placée dans le PC (eth1 dans notre exemple)
- Vérifiez quand même qu'il n'y ait qu'une seule passerelle de sortie …

Conclusion

- Notre serveur Linux sert donc maintenant de passerelle vers Internet, tout étant serveur DHCP et DNS
- Ces rôles auraient pu être tenus par un routeur multifonction, mais notre serveur pourra encore tenir d'autres rôles, comme celui de serveur de fichiers, ou de contrôleur de domaine …
- Merci de votre attention