Cours 5 : Réseaux privés NAT

Rabii El Ghorfi

Plan

1. Introduction

2. NAT statique

3. NAT dynamique : Masquerading

4. Proxy

Plan

1. Introduction

2. NAT statique

3. NAT dynamique : Masquerading

4. Proxy

Pourquoi avoir des adresses privées?

- Gérer la pénurie d'adresses au sein d'un réseau
- Masquer l'intérieur du réseau par rapport à l'extérieur (le réseau peut être vu comme une seule et même machine)
- Améliorer la sécurité pour le réseau interne
- Assouplir la gestion des adresses du réseau interne
- Faciliter la modification de l'architecture du réseau interne

$\rightarrow$ Mécanisme de translation d'adresses (NAT - Network Address Translation)
Deux types de NAT :

statique. association entre *n* adresses publiques et *n* adresses privées.

dynamique. association entre 1 adresse publique et *n* adresses privées.

Plan

1. Introduction

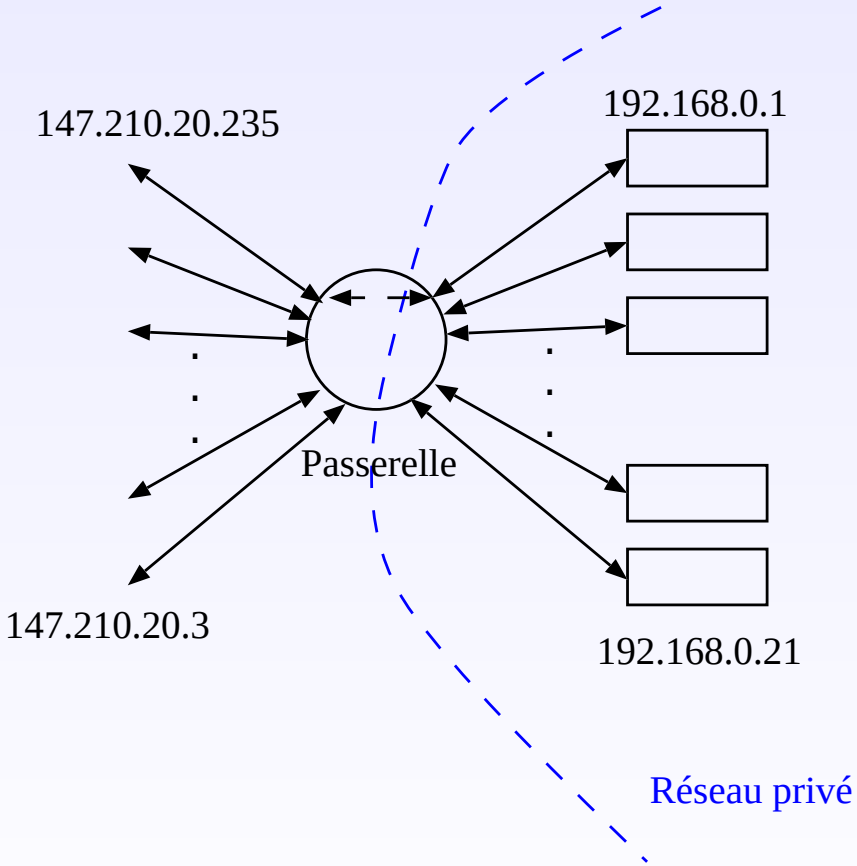
2. NAT statique

3. NAT dynamique : Masquerading

4. Proxy

NAT statique

Association entre une adresse publique et une adresse privée.



NAT statique

Association entre une adresse publique et une adresse privée.

Intérêt :

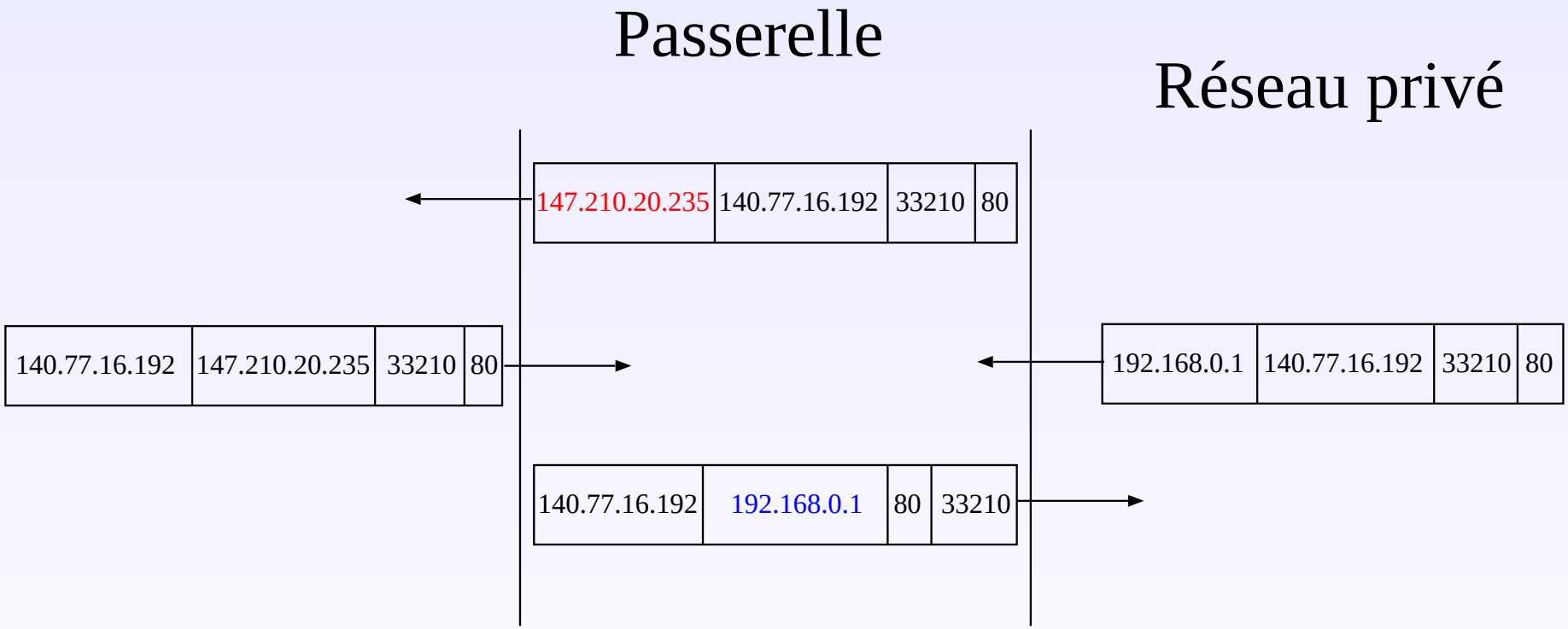
- Uniformité de l'adressage dans la partie privée du réseau (modification de la correspondance @publique/@privée facile)
- Sécurité accrue (tous les flux passent par la passerelle NAT)

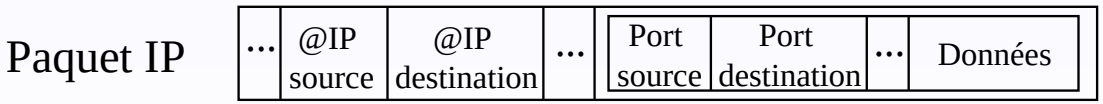
Inconvénient :

- Problème de pénurie d'adresses IP publiques non-résolu

NAT statique : Principe

Pour chaque paquet sortant (resp. entrant), la passerelle modifie l'adresse source (resp. destination).





Plan

1. Introduction

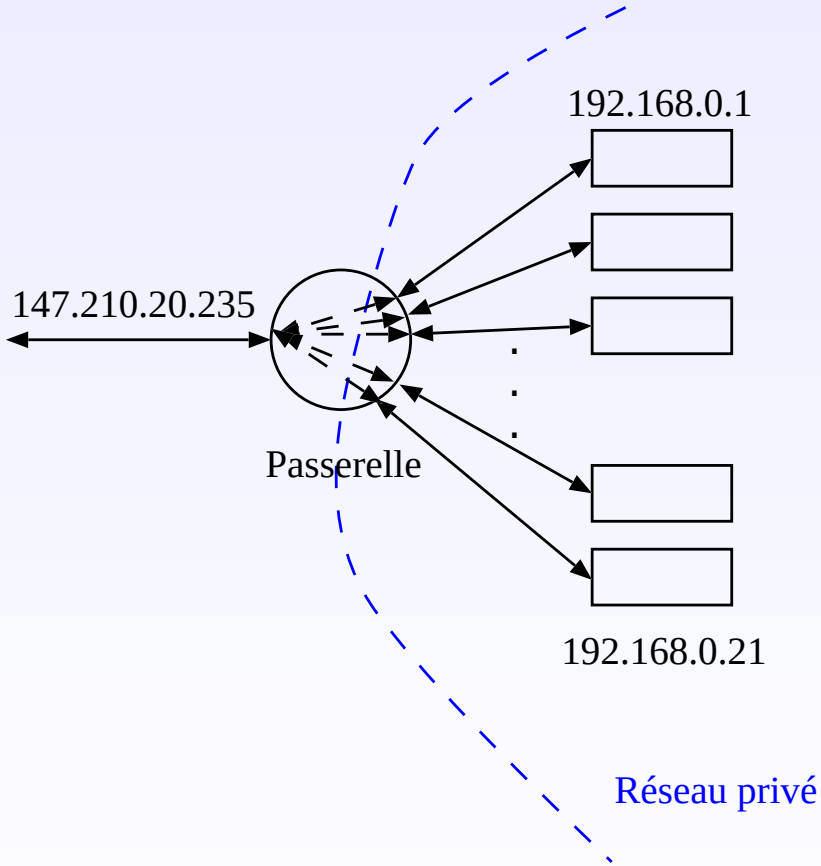
2. NAT statique

3. NAT dynamique : Masquerading

4. Proxy

NAT dynamique : Masquerading

Association entre m adresses publiques et n adresses privées ($m < n$).



NAT dynamique : Masquerading

Association entre m adresses publiques et n adresses privées ($m < n$).

Intérêt :

- Plusieurs machines utilisent la même adresse IP publique pour sortir du réseau privé
- Sécurité accrue (tous les flux passent par la passerelle NAT)

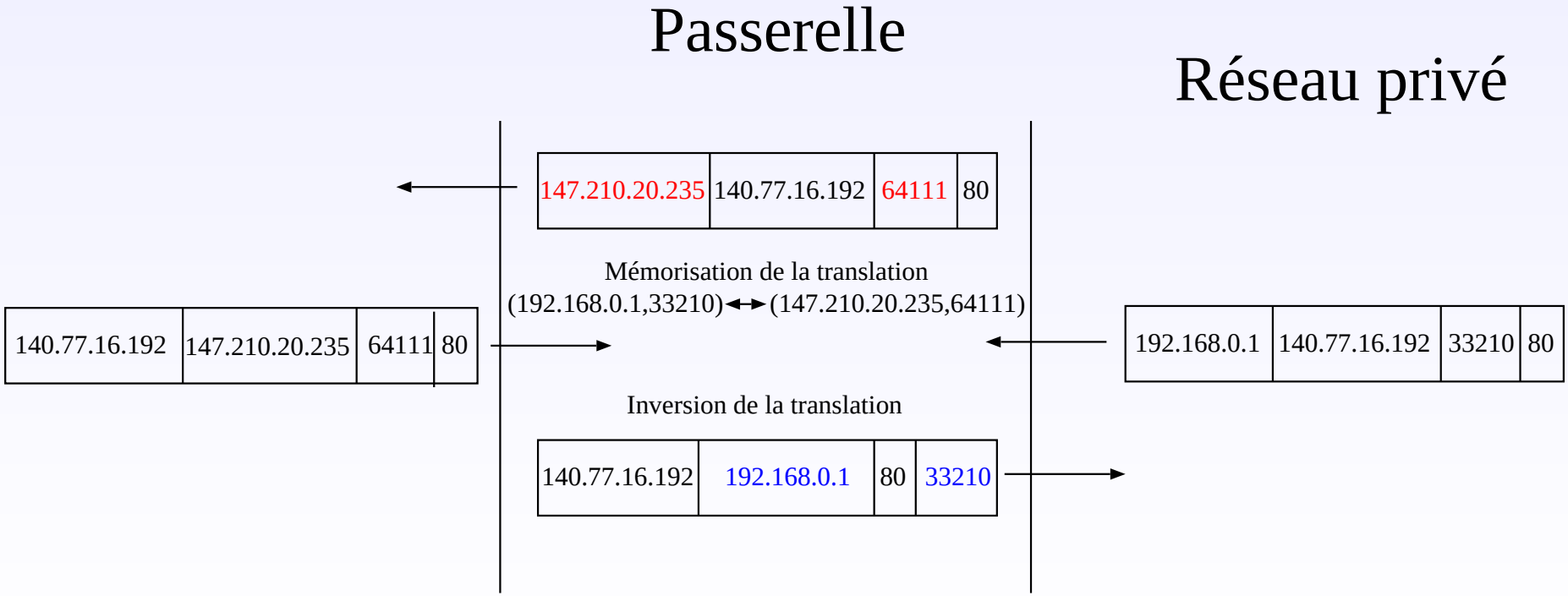
Inconvénient :

- Les machines du réseau interne ne sont pas accessibles de l'extérieur (impossibilité d'initier une connexion de l'extérieur)

NAT dynamique : Principe (1/2)

L'association de *n* adresses privées à 1 adresse publique nécessite, au niveau de la passerelle, de :

- ★ modifier l'adresse source (resp. destination) des paquets sortant (resp. entrants)
- ★ changer le numéro de port source pour les flux sortant



NAT dynamique : Principe (2/2)

Comment est ce que le routeur différencie les paquets qui lui sont destinés de ceux qu'il doit relayer?

À chaque nouvelle connexion :

1: Modifier l'adresse source et le port source : (@source_privée,port_source)→(@publique,port_source')
2: Sauvegarder l'association dans la table `NAT`

Pour chaque paquet entrant :

3: Chercher une association correspondant au couple (@destination, port_destination)
4: **Si** ∃ une association dans la table `NAT` **Alors**
5: Modifier l'adresse de destination et le port de destination
6: Relayer le paquet
7: **Sinon**
8: /* Erreur de routage */
9: **Fin du Si**

NAT dynamique : Principe (2/2)

Comment est ce que le routeur différencie les paquets qui lui sont destinés de ceux qu'il doit relayer?

À chaque nouvelle connexion :

1: Modifier l'adresse source et le port source : (@source_privée,port_source)→(@publique,port_source')
2: Sauvegarder l'association dans la table `NAT`

Pour chaque paquet entrant :

3: Chercher une association correspondant au couple (@destination, port_destination)
4: **Si** ∃ une association dans la table `NAT` **Alors**
5: Modifier l'adresse de destination et le port de destination
6: Relayer le paquet
7: **Sinon**
8: /* Erreur de routage */
9: **Fin du Si**

Le routeur gère toutes les associations

⇒ Unicité de l'association (donc du port source après translation)

Problèmes liés à NAT dynamique

Comment faire de la translation d'adresse sur des protocoles qui ne sont pas basés sur TCP ou UDP (pas de numéro de port)?

- Nécessité d'implémenter une méthode spécifique au protocole (identifiant ICMP pour ICMP par exemple).
- Dans le cas des protocoles dont les paquets contiennent des données relatives aux adresses IP, il est nécessaire de mettre en place des "proxy" (`FTP` en mode actif par exemple).

Comment rendre joignables des machines du réseau local?

- Nécessité de faire de la redirection de port (port forwarding/mapping).

  Principe. Toutes les connexions entrantes sur un port donné sont redirigée vers une machine du réseau privé sur un port (qui peut être le même ou non).

Plan

1. Introduction

2. NAT statique

3. NAT dynamique : Masquerading

4. Proxy

Proxy ou mandataire

Définition :

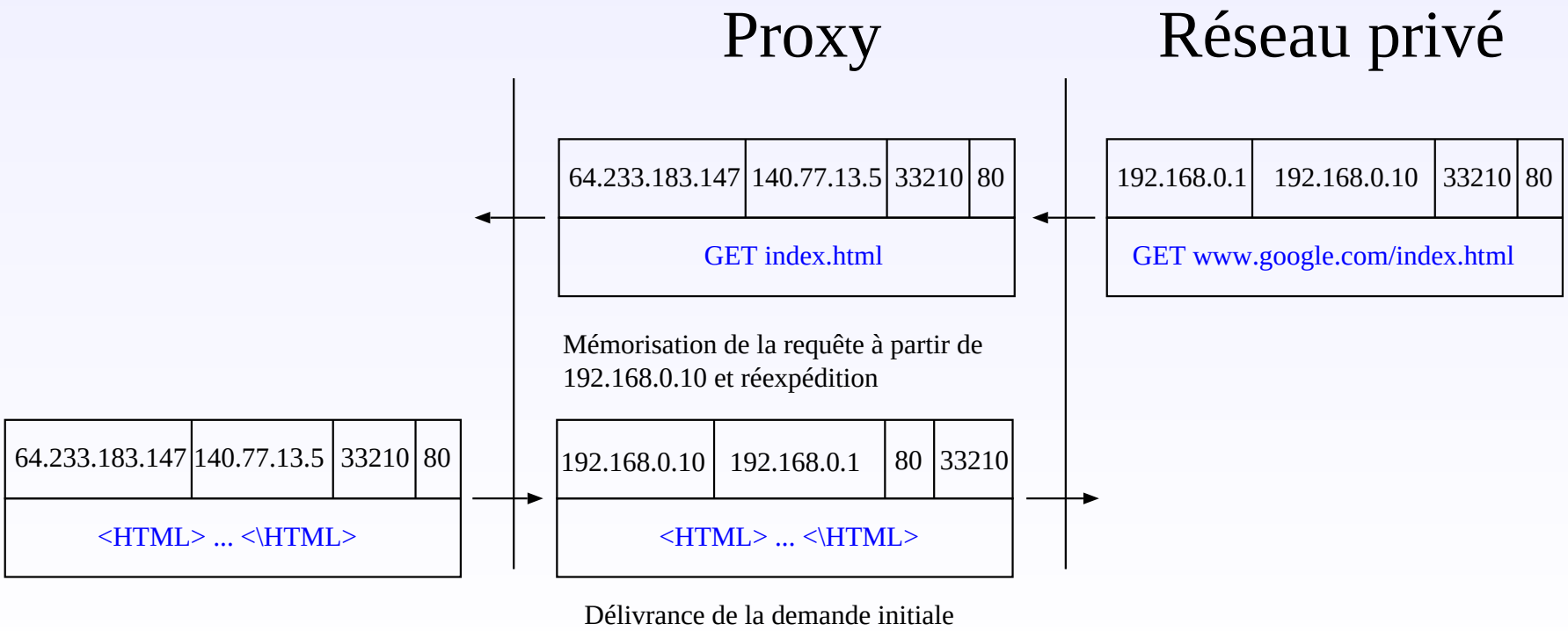
- Un proxy est un intermédiaire dans une connexion entre le client et le serveur
- Le client s'adresse toujours au proxy
- Le proxy est spécifique à une application donnée (`HTTP`, `FTP`, ...)

$\rightarrow$ Possibilité de modification des informations échangées entre le client et le serveur.

Proxy ou mandataire

Définition :

- Un proxy est un intermédiaire dans une connexion entre le client et le serveur
- Le client s'adresse toujours au proxy
- Le proxy est spécifique à une application donnée (`HTTP`, `FTP`, ...)



Réseaux privés Partie 2 : NAT et commandes Iptable

Plan

1. Logiciels de filtrage de paquets
2. Ipfwadm
3. Ipchains
4. Iptables

Plan

1. Logiciels de filtrage de paquets

2. Ipfwadm

3. Ipchains

4. Iptables

Logiciels de filtrage de paquets

- Fonctionnalités de “firewall” filtrant directement implémentée dans le noyau Linux.
- Filtrage de niveau 3 ou 4.
- 3 types de firewall filtrants :

  Ipfwadm. Jusqu’à la version 2.1.102 du noyau linux

  Ipchains. Entre les versions 2.2.0 et 2.4 du noyau linux

  Iptables. À partir des noyaux 2.4

Plan

1. Logiciels de filtrage de paquets

2. Ipfwadm

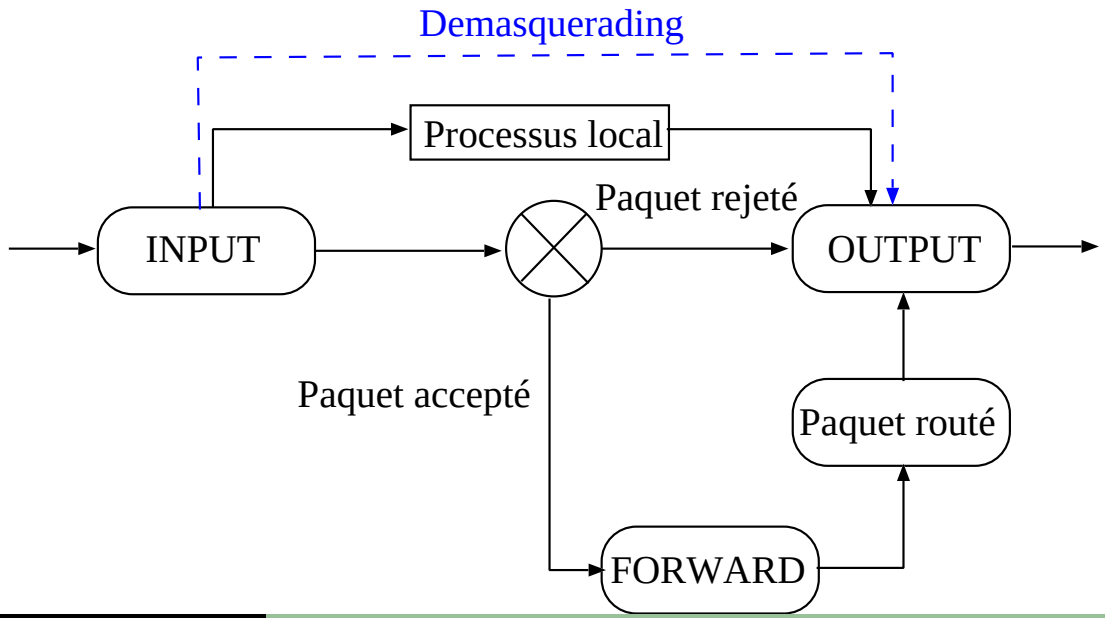
3. Ipchains

4. Iptables

Ipfwadm

- Firewall permettant la gestion des paquets TCP, UDP et ICMP.
- 3 types de règles :
  - **INPUT.** sont appliquées lors de l'arrivée d'un paquet.
  - **FORWARD.** sont appliquées lorsque la destination du paquet n'est pas le routeur.
  - **OUTPUT.** sont appliquées dès qu'un paquet doit sortir du routeur.

Fonctionnement :



Ipfwadm

- Firewall permettant la gestion des paquets TCP, UDP et ICMP.
- 3 types de règles :
  - INPUT. sont appliquées lors de l'arrivée d'un paquet.
  - FORWARD. sont appliquées lorsque la destination du paquet n'est pas le routeur.
  - OUTPUT. sont appliquées dès qu'un paquet doit sortir du routeur.

Fonctionnement :

1: lorsqu'un paquet entre, il traverse les règles de type INPUT
2: **Si** il est accepté **Alors**
3: **Si** il est destiné à une autre machine **Alors**
4: il est routé vers les règles FORWARD
5: **Sinon**
6: il est rejeté
7: le paquet est finalement émis

Dans tous les cas, le paquet traverse les règles OUTPUT

Plan

1. Logiciels de filtrage de paquets

2. Ipfwadm

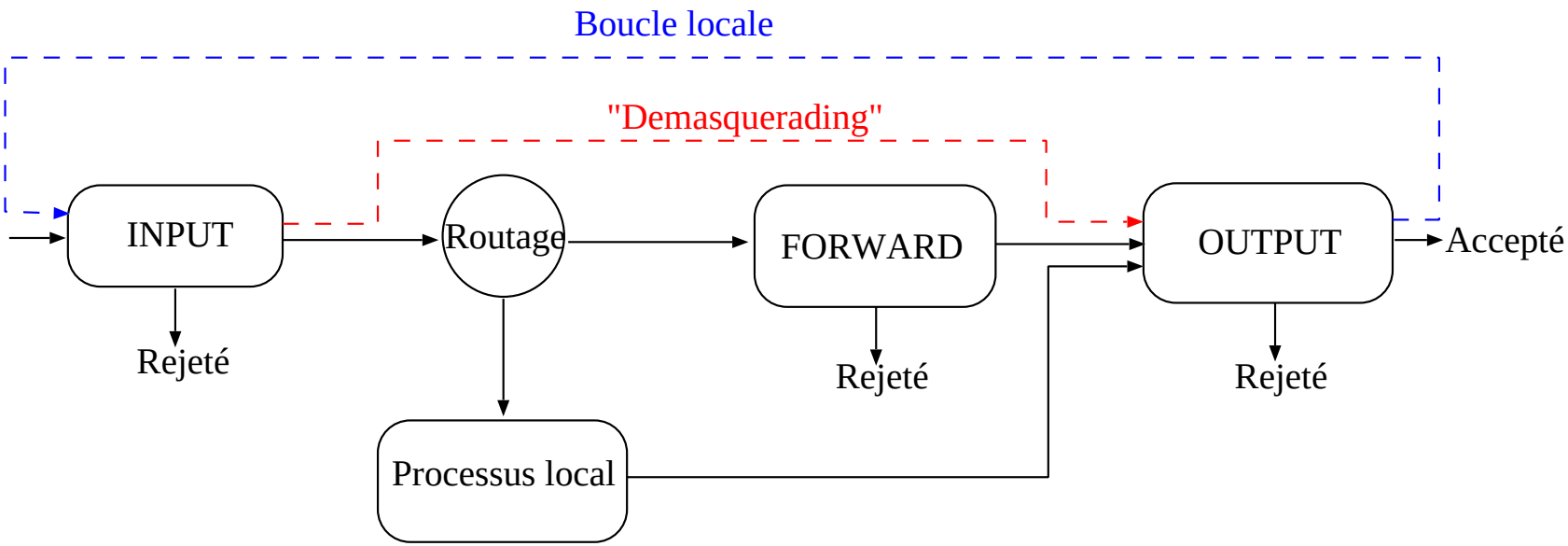
3. Ipchains

4. Iptables

Ipchains

- Module du noyau Linux réalisant le filtrage de paquets.
- Inspiré du parre-feu BSD (tout comme ipfwadm)

Fonctionnement :



Plan

1. Logiciels de filtrage de paquets

2. Ipfwadm

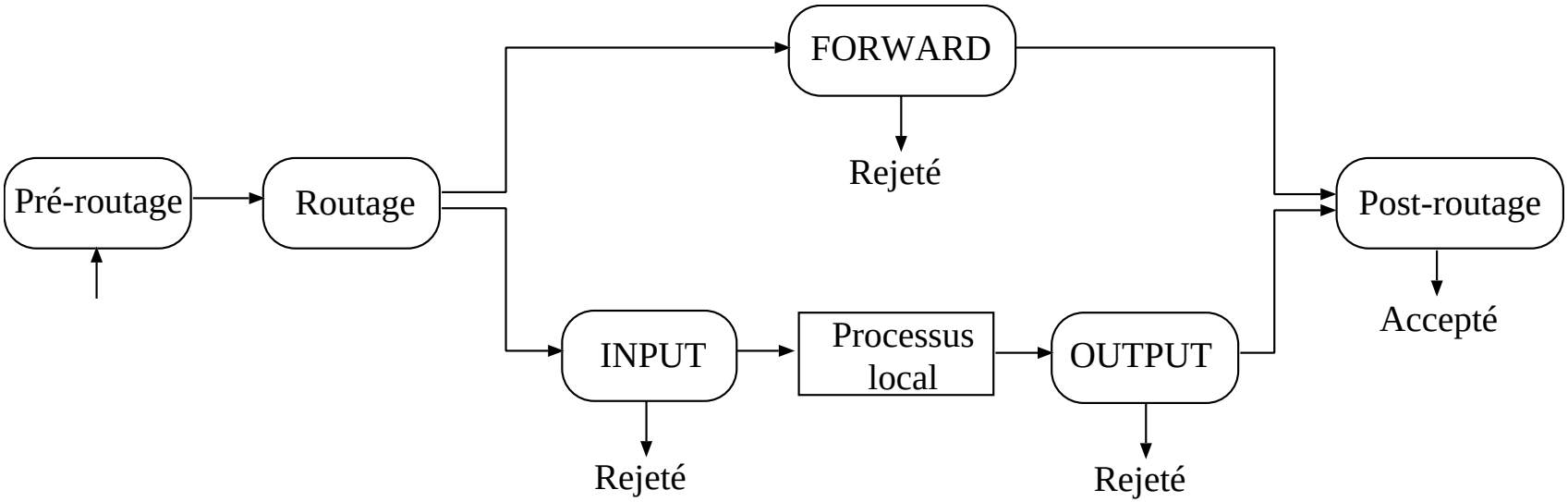
3. Ipchains

4. Iptables

Iptables (1/2)

- Module du noyau Linux réalisant le filtrage de paquets (noyaux $\geq$ 2.4).
- Améliorations en matière de filtrage et de translation d'adresses par rapport à Ipchains.

Fonctionnement :



Iptables (1/2)

- Module du noyau Linux réalisant le filtrage de paquets (noyaux $\geq$ 2.4).
- Améliorations en matière de filtrage et de translation d'adresses par rapport à Ipchains.

Fonctionnement :

À l'arrivée d'un paquet (après décision de routage) :

Si le paquet est destiné à l'hôte local **Alors**
    il traverse la chaîne INPUT.
    Si il n'est pas rejeté **Alors**
        il est transmis au processus impliqué.
Sinon
    Si le paquet est destiné à un hôte d'un autre réseau **Alors**
        il traverse la chaîne FORWARD
        Si il n'est pas rejeté **Alors**
            il poursuit alors sa route

Iptables (1/2)

- Module du noyau Linux réalisant le filtrage de paquets (noyaux $\geq$ 2.4).
- Améliorations en matière de filtrage et de translation d'adresses par rapport à Ipchains.

Fonctionnement :

À l'arrivée d'un paquet (après décision de routage) :

Si le paquet est destiné à l'hôte local **Alors**
    il traverse la chaîne INPUT.
    Si il n'est pas rejeté **Alors**
        il est transmis au processus impliqué.
Sinon
    Si le paquet est destiné à un hôte d'un autre réseau **Alors**
        il traverse la chaîne FORWARD
        Si il n'est pas rejeté **Alors**
            il poursuit alors sa route

Tous les paquets émis par des processus locaux au routeur traversent la chaîne OUTPUT.

Iptables (2/2)

Fonctionnalités :

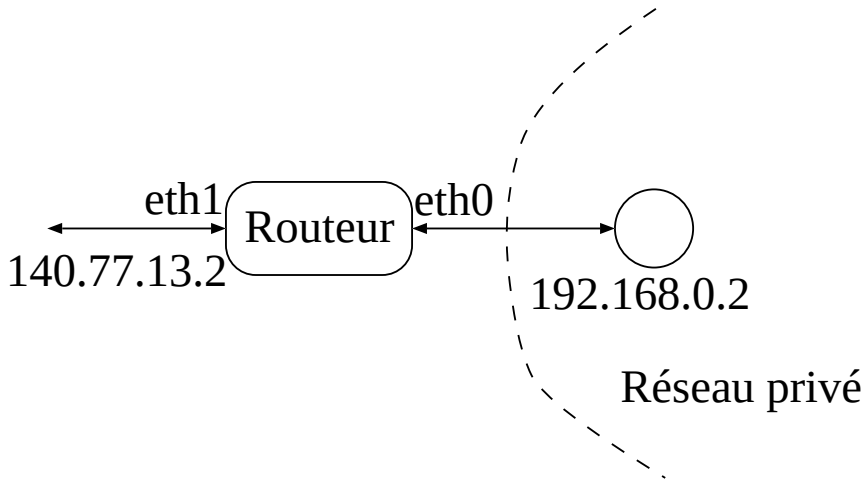
- Filtrage de paquets
- NAT
- Marquage de paquets

Architectures :Trois tables de chaînes (FILTER, NAT et MANGLE).

FILTER (filtrage des paquets)		NAT (translation d'adresses)	
INPUT	paquet entrant sur le routeur	PREROUTING	NAT de destination
OUTPUT	paquet émis par le routeur	POSTROUTING	NAT de source
FORWARD	paquet traversant le routeur	OUTPUT	NAT sur les paquets émis localement

La table MANGLE sert au marquage des paquets

Fonctionnalités NAT d'Iptables (1/2)



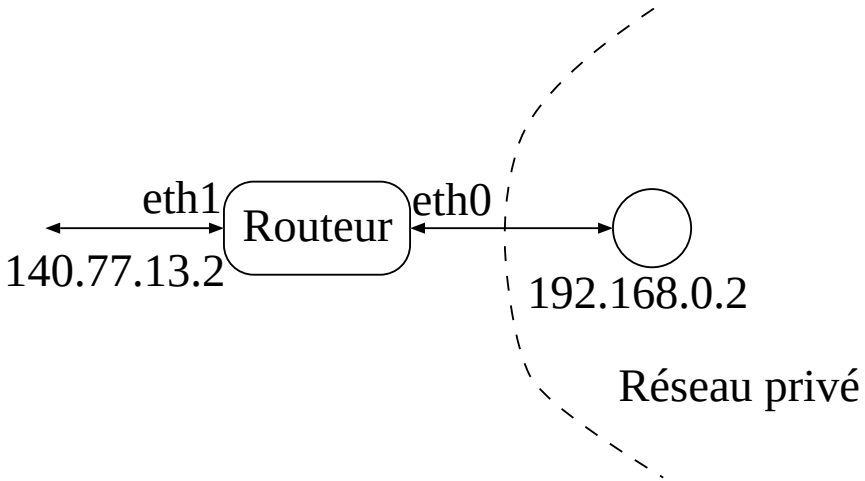
Modification de la destination du paquet avant le routage (paquet reçu de l'extérieur).

```
iptables -t nat -A PREROUTING -d 140.77.13.2 -i eth1 -j DNAT
--to-destination 192.168.0.2
```

Modification de la source du paquet après le routage (paquet émis à partir du réseau privé).

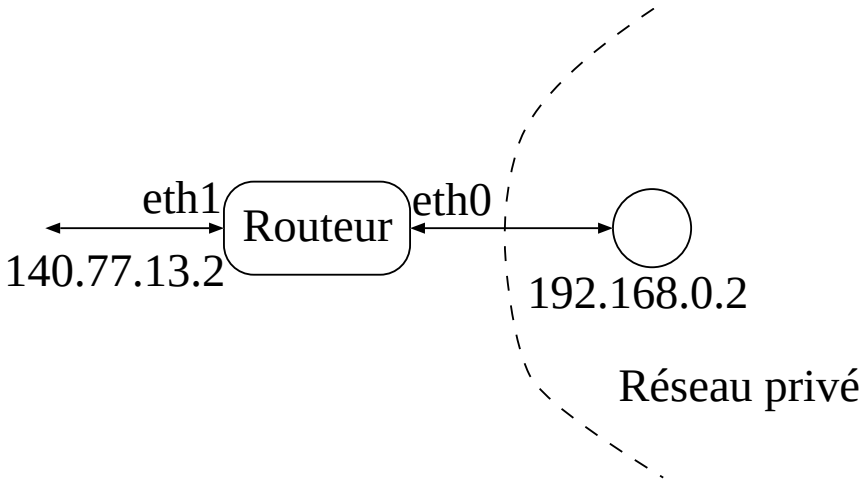
```
iptables -t nat -A POSTROUTING -s 192.168.0.2 -o eth1 -j
SNAT --to-source 140.77.13.2
```

Fonctionnalités NAT d'Iptables (1/2)



Exercice : Comment faire pour que le routeur puisse envoyer un paquet à l'adresse 140.77.13.2?

Fonctionnalités NAT d'Iptables (1/2)



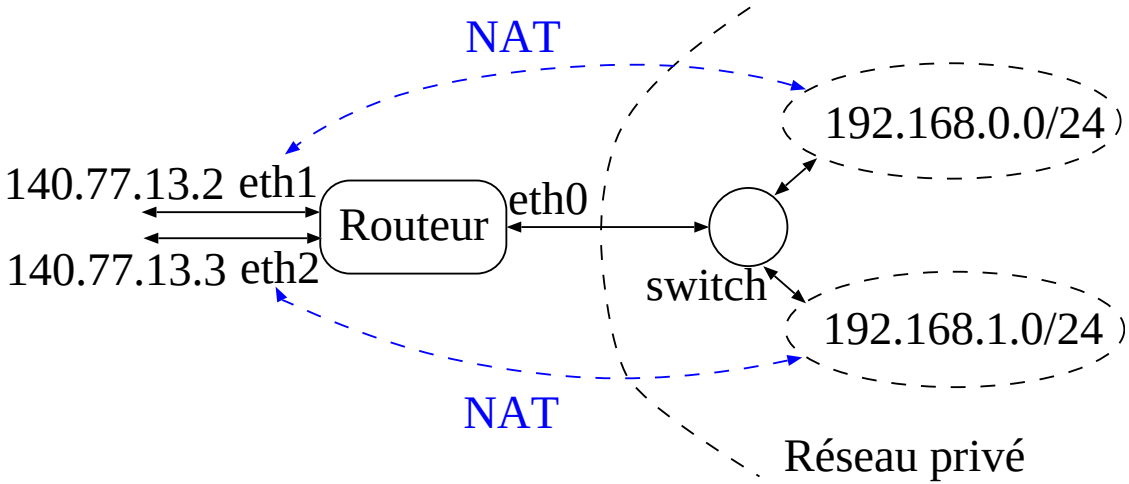
Exercice : Comment faire pour que le routeur puisse envoyer un paquet à l'adresse 140.77.13.2?

Réponse :

Il faut modifier la destination du paquet émis localement avant le routage.

```
iptables -t nat -A OUTPUT -d 140.77.13.2  -j DNAT
--to-destination 192.168.0.2
```

Fonctionnalités NAT d'Iptables (2/2)



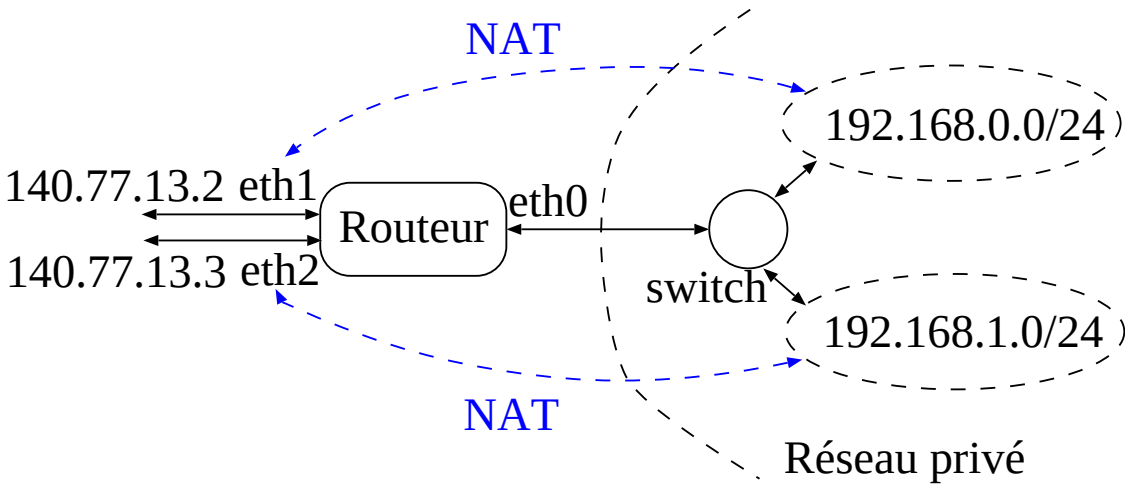
Association entre toutes les adresses privées du sous-réseau 192.168.0.0/24 avec l'interface eth1.

```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.0/24 -j
MASQUERADE
```

Association entre toutes les adresses privées du sous-réseau 192.168.1.0/24 avec l'interface eth2.

```
iptables -t nat -A POSTROUTING -o eth2 -s 192.168.1.0/24 -j
MASQUERADE
```

Transfert de ports



Transférer les connexions sur le port 80 de l'adresse 140.77.13.2 sur la machine ayant l'adresse privée 192.168.0.200 sur le port 8080 :

```
iptables -t nat -A PREROUTING -p tcp -d 140.77.13.2 --dport
80 --sport 1024:65535 -j DNAT --to 192.168.0.200:8080
```