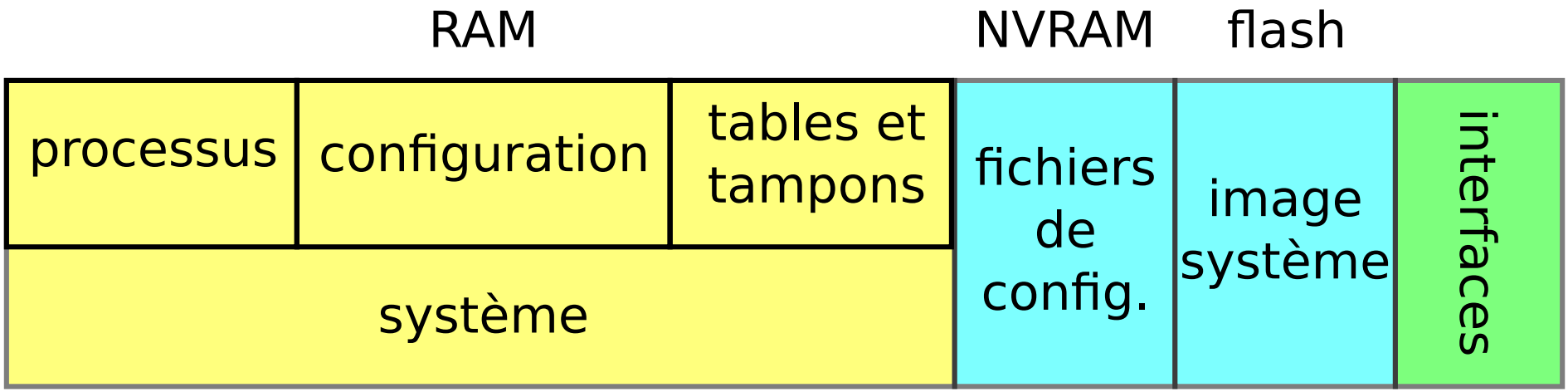
Configuration des matériels Cisco

Yvan Peter

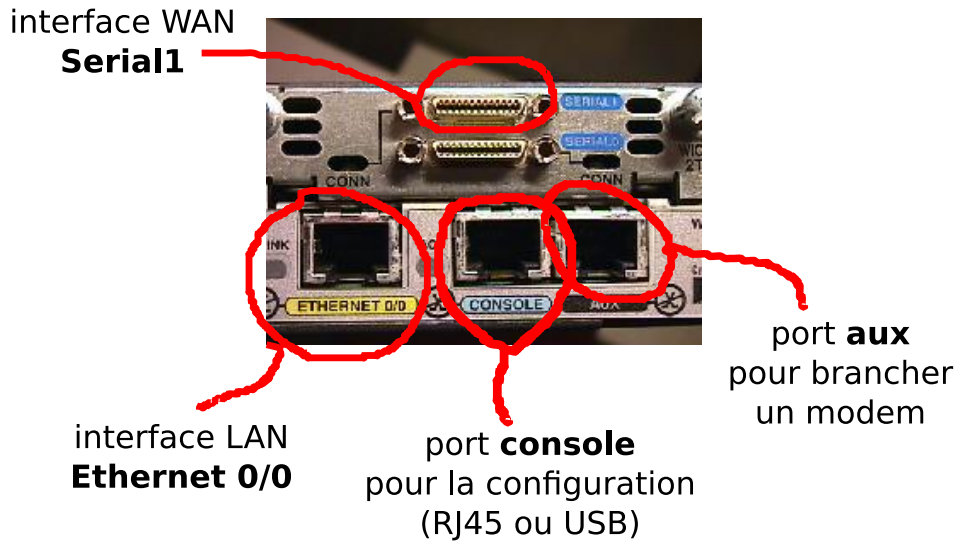
IUT A - Université de Lille

Architecture du routeur



Interfaces et connexion

- Interface série via le port console avec un émulateur de terminal (`minicom`, `Putty`...)
  configuration : 9600 bits/s, pas de contrôle de flux, 8 bits de données, pas de parité, 1 bit de stop (8N1)
- telnet / ssh
  - IP doit être configuré
  - l'accès doit être protégé par un mot de passe
- serveur web interne



Système d'exploitation

Internet Operating System (IOS)

- Le système historique

IOS XR / XE, NX-OS

- basé sur un noyau Linux (CentOS)
- plus sûr (processus séparés)
- extensible (containers)
- API de gestion

Modes de configuration

Exec utilisateur
examen limité
du routeur
`Router>`

Exec privilégié
examen détaillé
du routeur
débogage et tests
manipulation des fichiers
de configuration
`Router#`

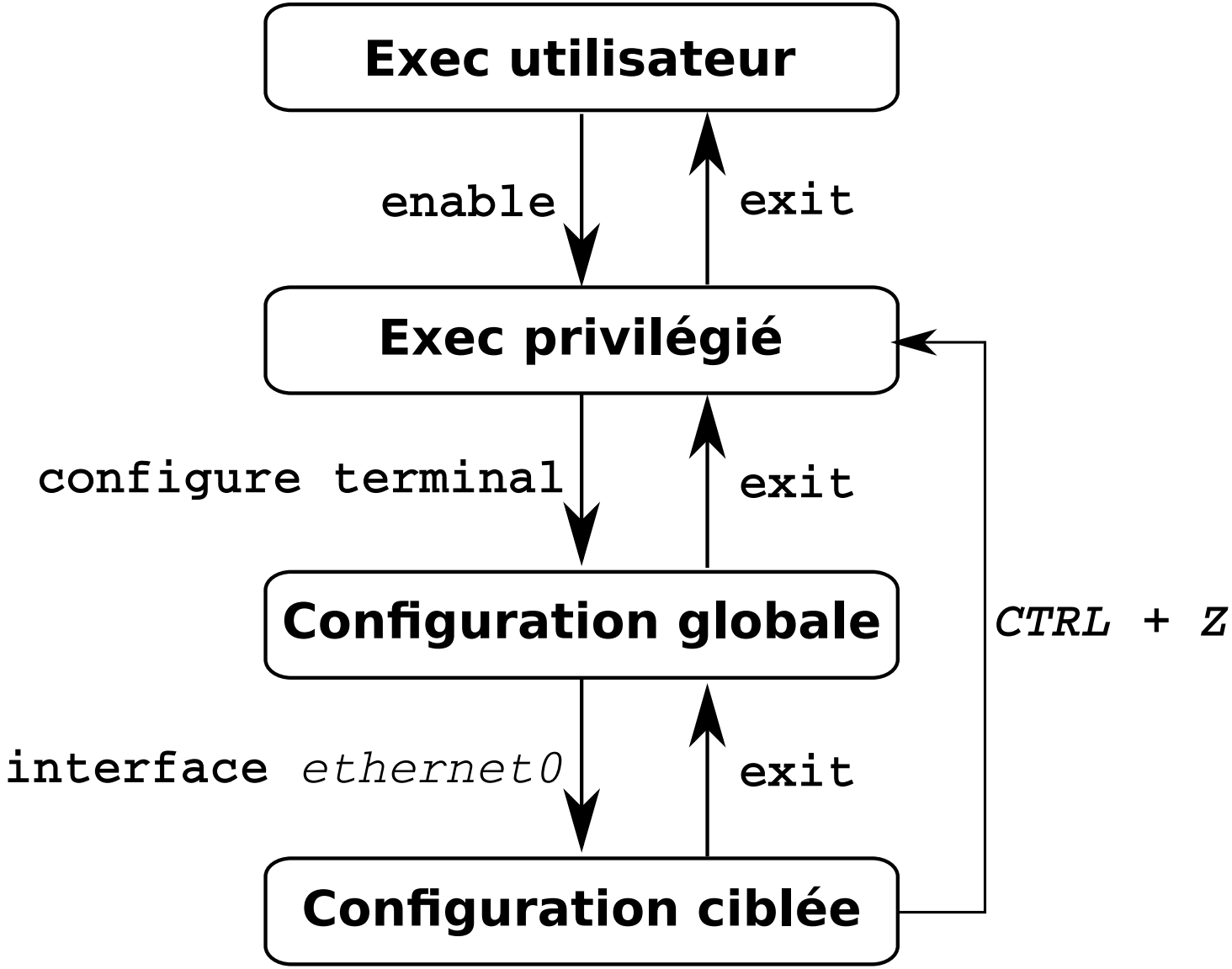
Configuration globale
gestion des paramètres
globaux du routeur
`Router<config>#`

Configuration ciblée
configuration de paramètes
spécifiques (interface, routage...)
`Router<config-mode>#`

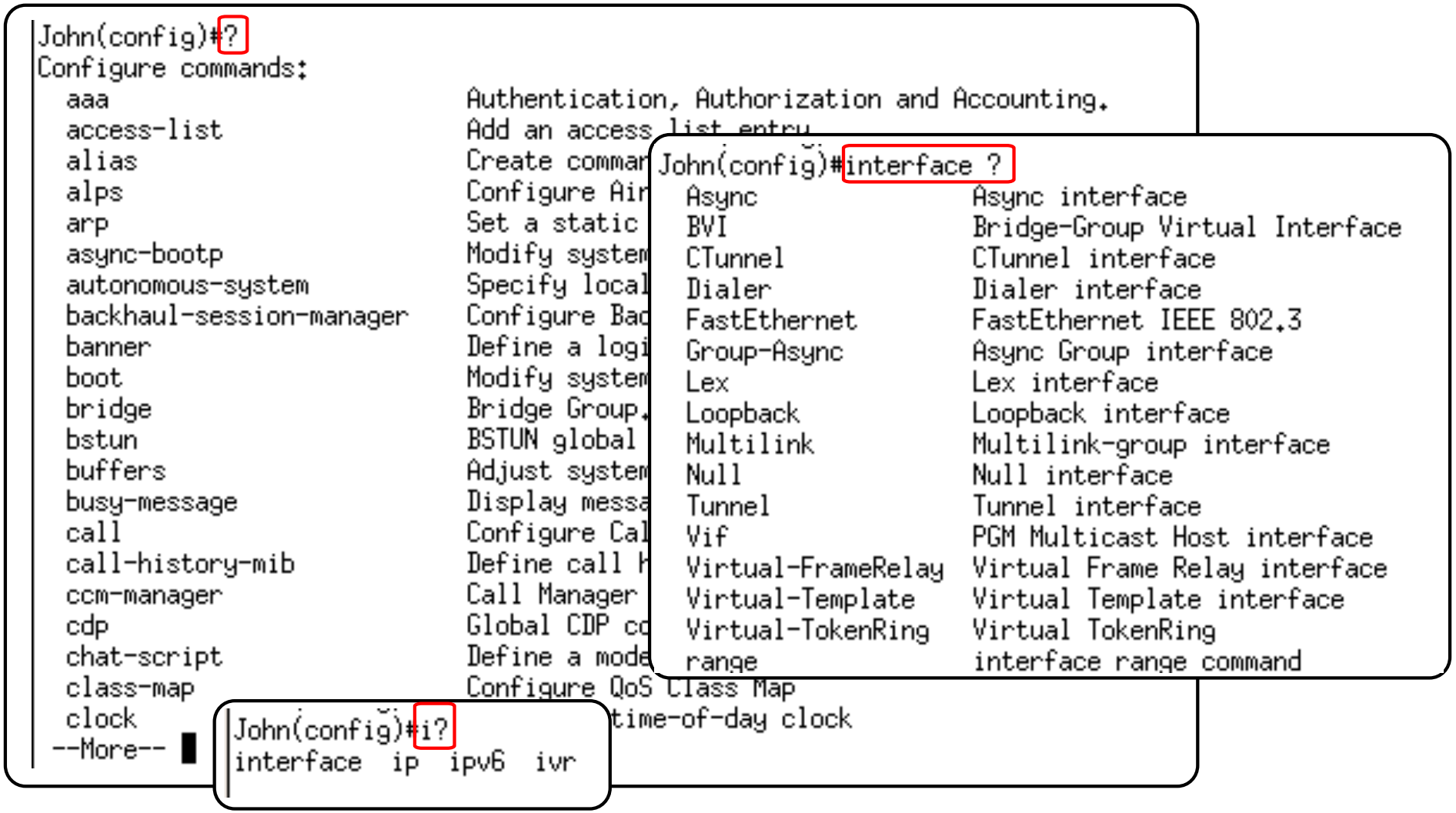
Dialogue de configuration initiale
Configuration guidée du routeur

Contrôle en ROM
Pas d'IOS trouvé ou altération
de la séquence de boot
`rommon>`

Modes de configuration



Obtenir de l'aide



Gestion des configurations

`startup-config`

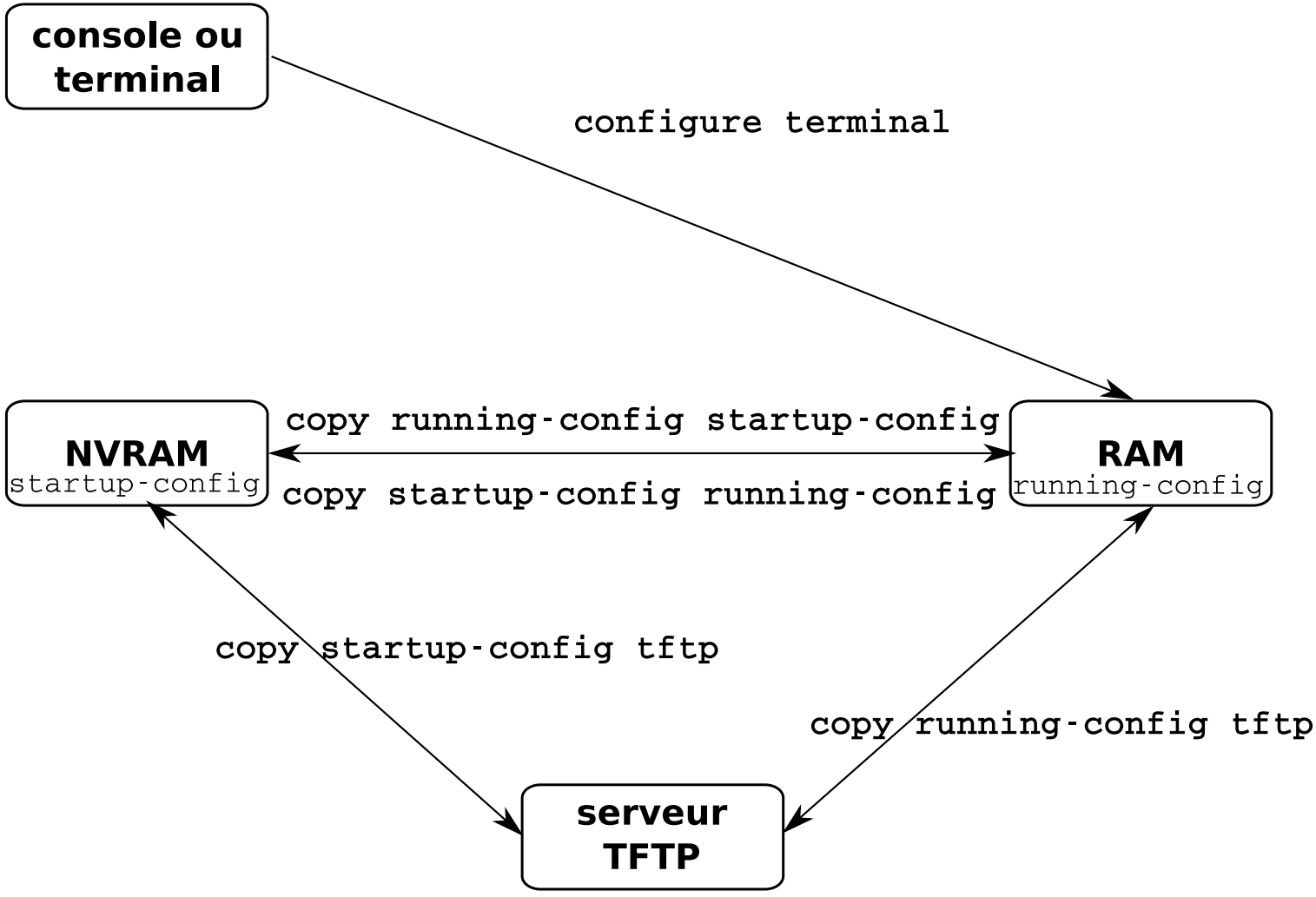
C'est la configuration chargée au démarrage du matériel

`running-config`

C'est la configuration en cours d'utilisation

toute modification est prise en compte immédiatement

Gestion des configurations



Configuration de base

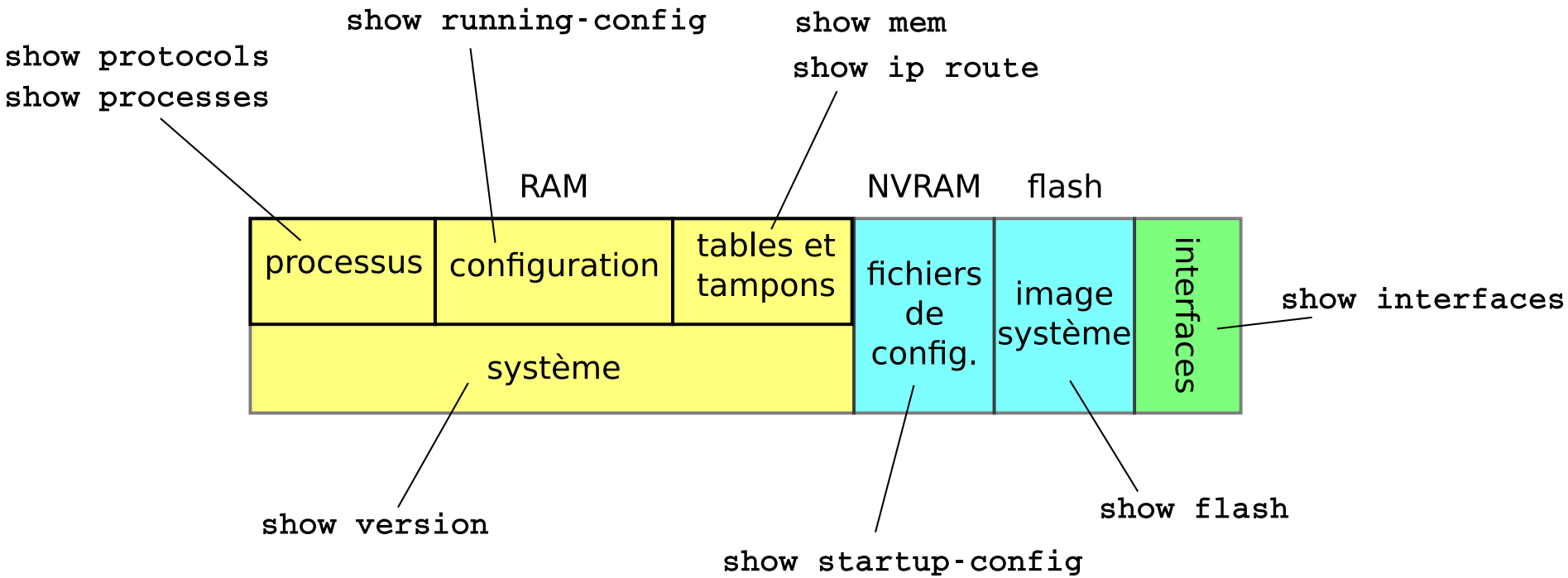
```
Current configuration : 649 bytes
!
version 12.2
no parser cache
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname John
!
logging rate-limit console 10 except errors
!
!
!
memory-size iomem 15
ip subnet-zero
!
!
!
no ip dhcp-client network-discovery
call rsvp-sync
 --More-- 
```

```
!
!
interface FastEthernet0/0
 ip address 192.168.5.1 255.255.255.0
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
ip classless
ip http server
!
!
!
!
dial-peer cor custom
!
!
 --More-- 
```

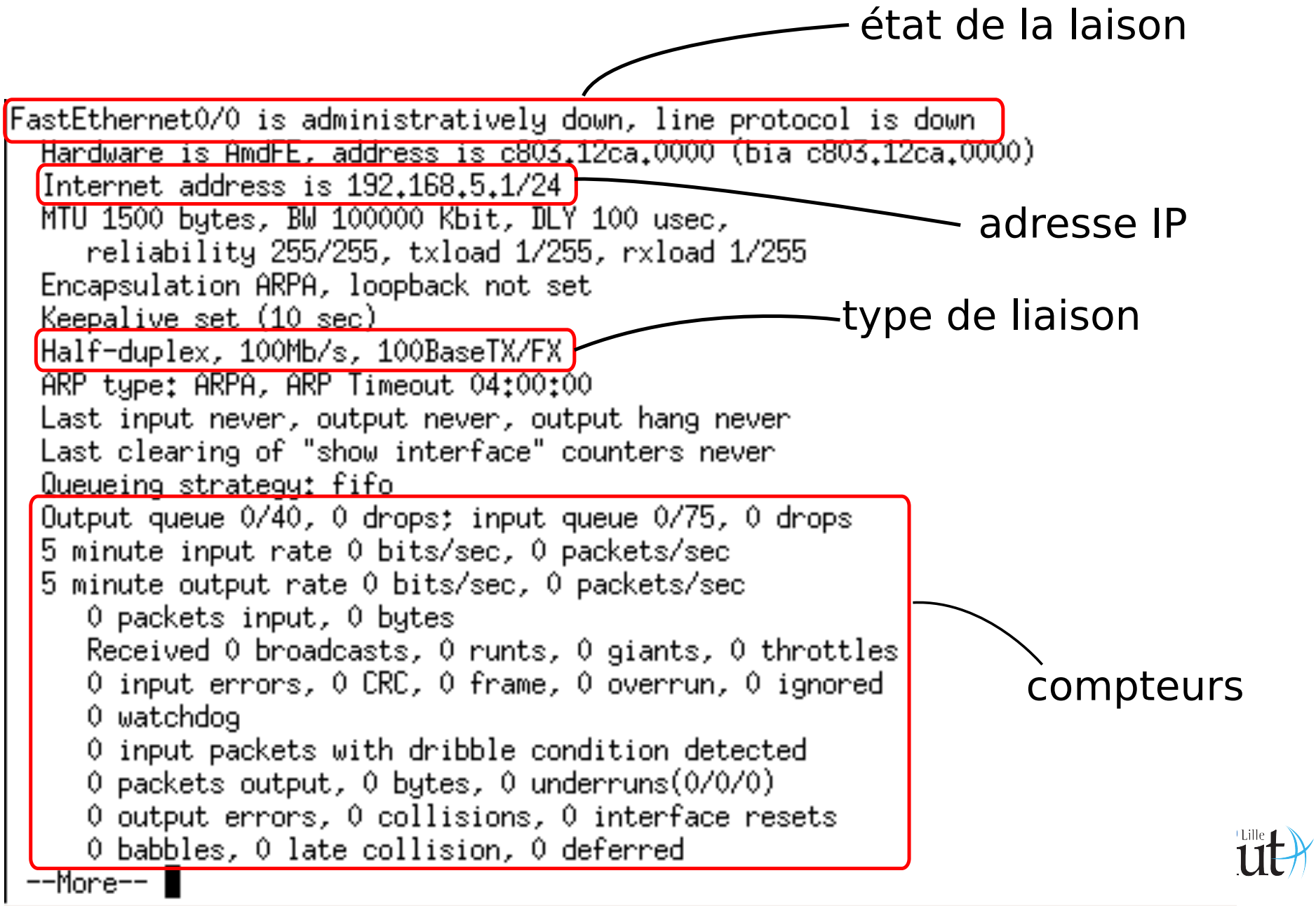
```
!
!
line con 0
line aux 0
line vty 5 15
!
no scheduler allocate
!
end

John#
```

La commande `show`



show interfaces



show interfaces

```
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is c803.12ca.0000 (bia c803.12ca.0000)
  Internet address is 192.168.5.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Half-duplex, 100Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:09, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     21 packets input, 3477 bytes
     Received 6 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
     41 packets output, 4956 bytes, 0 underruns(0/0/0)
     0 output errors, 0 collisions, 2 interface resets
     0 babbles, 0 late collision, 0 deferred
```

show version

```
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-IS-M), Version 12.2(2)T1,  RELEASE SOFTWARE (fc2)
TAC Support: http://www.cisco.com/tac
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Wed 18-Jul-01 08:23 by ccai
Image text-base: 0x80008088, data-base: 0x8118CD90
```

La commande `debug`

- la commande `debug` permet d'obtenir des logs sur la console.
- le debogage affecte les performances (cibler les traces)
- par défaut uniquement en console
  - utiliser `terminal monitor` pour obtenir les traces dans un `vty`

Exemple de configuration

```
Router> enable
Router# configure terminal
Router(config)# hostname r1
r1(config)# banner motd # service info #
r1(config)# enable secret cgir
r1(config)# service password-encryption
r1(config)# exit
r1# copy running-config startup-config
```

Configuration du nom et du message d'accueil

Gestion du mot de passe exec privilégié

Sauvegarde de la configuration

Sécuriser l'accès au routeur

Exec privilégié

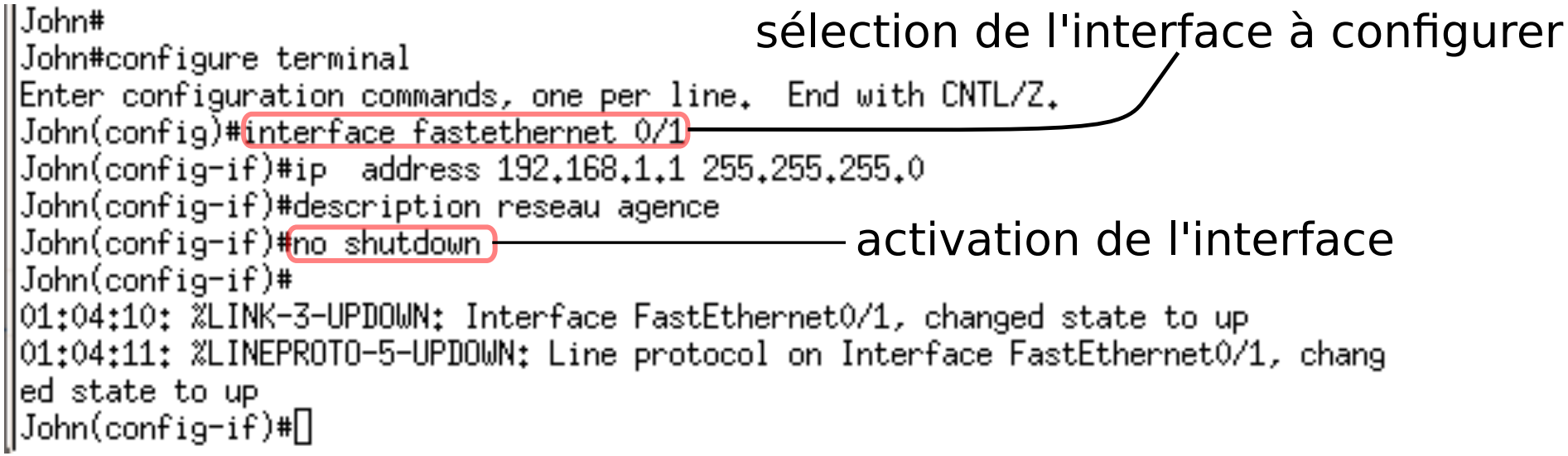
`enable secret` permet de définir le mot de passe

`service password-encryption` permet (juste) de le rendre illisible

Accès Console et vty (telnet/ssh)

```
line con 0                    ← console
  password le_mdp
  login                       ← activation du mot de passe
  exec-timeout minutes [secondes]
line vty 0 4                  ← terminaux virtuels
  password le_mdp
  login
  exec-timeout minutes [secondes]
                              ← gestion de la déconnexion
```

Configuration d'une interface



Nommage des interfaces

Les interfaces sont nommées selon la technologie (Ethernet, FastEthernet...Serial, etc.), le numéro de la carte d'interface puis le numéro de l'interface sur la carte

ex. `FastEthernet 0/1`

Routage statique

ip route `ip_réseau_dest [masque] ad_rout_suivant | nom_interface [`**permanent**`]`

- on peut donner l'adresse du routeur suivant ou l'interface de sortie vers ce routeur
- `permanent` conserve l'adresse dans la table de routage même si le lien n'est pas actif

Routage dynamique : RIP

RIP

```
router rip
  version 1|2
  network numero_reseau
  ...
  passive-interface interface
```

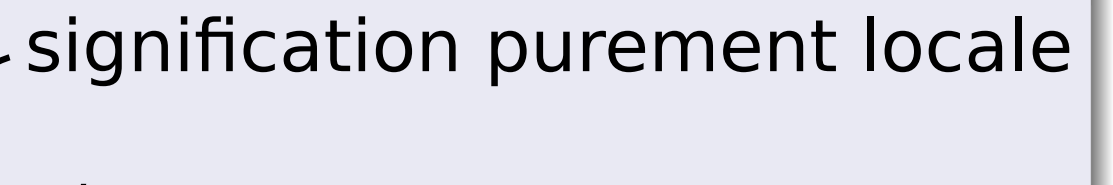
le réseau participe au routage (= envoi & réception des vecteurs de distance)

inhibe les envois de vecteur de distance

- les réseaux inclus dans la configuration sont ceux **qui sont physiquement connectés au routeur et que l'on veut inclure dans le routage**
- rappel : RIPv1 est *classful*

Routage dynamique : OSPF

OSPF




```
router ospf process_id
  network reseau wildcard area numéro_aire
  ...
```

- *wildcard* = «inverse du masque», indique la partie numéro de réseau

Routage dynamique : OSPF

Choix du *router-id*

- le *router-id* est utilisé pour déterminer le routeur désigné
- on configure l'interface de *loopback* pour fixer cet identifiant

```
interface loopback 0
  ip address réseau masque
```

Access Control List

ACL

- conditions sur le paquet et décision d'acceptation et de refus
- la première condition qui correspond est utilisée
- l'ordre des conditions est important

En pratique

- une ACL = numéro ou un nom communs pour un ensemble de conditions
- l'ACL doit être appliquée à une interface
- les ACL peuvent aussi être utilisées pour d'autres usages que le filtrage des paquets

Access Control List : différents types

ACL standard

- conditions sur l'adresse IP source uniquement
- numérotation : 1 – 99

ACL étendue

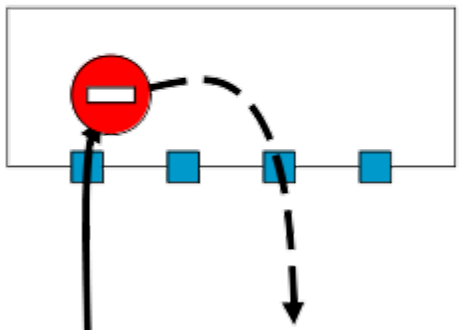
- conditions sur
  - l'adresse IP source et/ou destination
  - port source et/ou destination
  - autre éléments du paquet
- numérotation : 100 – 199

ACL nommée

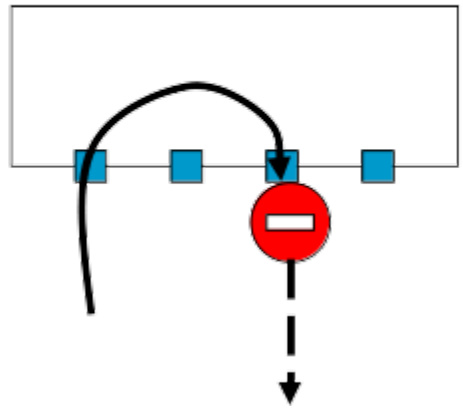
Peut être standard ou étendue. Offre des facilités d'édition.

Access Control List : application à une interface

Filtrage en **entrée** (`in`)



Filtrage en **sortie** (`out`)



ACL standard

access-list `num` {**deny** | **permit**} `source [wildcard]`

exemples

`access-list 1 permit 10.1.1.1`	autorise (permit) les paquets émis par la machine `10.1.1.1`
`access-list 1 permit` **host** `10.1.1.1`	syntaxe plus ancienne
`access-list 1 deny 192.168.5.0 0.0.0.255`	interdit (deny) tous les paquets venant du réseau `192.168.5.0/24`

ACL standard

```
access-list 42 permit 36.4.0.2
access-list 42 deny 36.4.0.0 0.0.255.255
access-list 42 permit 36.0.0.0 0.255.255.255
```

- paquet de source 36.5.0.3
- paquet de source 36.4.5.1
- paquet de source 36.4.0.2
- paquet de source 192.168.5.1

ACL standard

```
access-list 42 permit 36.4.0.2
access-list 42 deny 36.4.0.0 0.0.255.255
access-list 42 permit 36.0.0.0 0.255.255.255
```

- paquet de source 36.5.0.3
- paquet de source 36.4.5.1
- paquet de source 36.4.0.2
- paquet de source 192.168.5.1

ACL standard

```
access-list 42 permit 36.4.0.2
access-list 42 deny 36.4.0.0 0.0.255.255
access-list 42 permit 36.0.0.0 0.255.255.255
```

- paquet de source 36.5.0.3
- paquet de source 36.4.5.1
- paquet de source 36.4.0.2
- paquet de source 192.168.5.1

ACL standard

```
access-list 42 permit 36.4.0.2
access-list 42 deny 36.4.0.0 0.0.255.255
access-list 42 permit 36.0.0.0 0.255.255.255
```

- paquet de source 36.5.0.3
- paquet de source 36.4.5.1
- paquet de source 36.4.0.2
- paquet de source 192.168.5.1

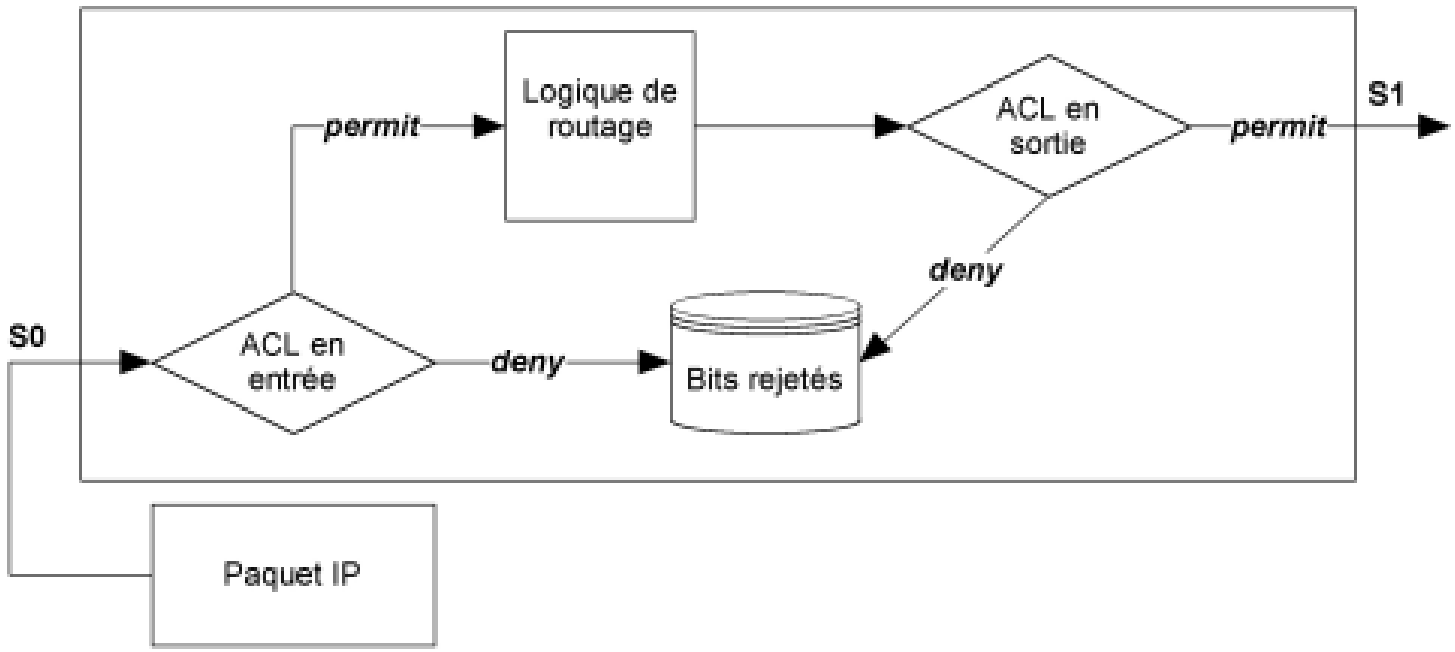
ACL standard

```
access-list 42 permit 36.4.0.2
access-list 42 deny 36.4.0.0 0.0.255.255
access-list 42 permit 36.0.0.0 0.255.255.255
access-list 42 deny any
```

- paquet de source 36.5.0.3
- paquet de source 36.4.5.1
- paquet de source 36.4.0.2
- paquet de source 192.168.5.1

ACL : application à une interface

ip access-group `num` **{in | out }**



ACL standard : principes d'utilisation

- planifier la localisation et la direction du filtrage
  - placer l'ACL standard proche de la destination pour éviter de supprimer des paquets inutilement
  - identifier les flux qui vont vers la destination
- configurer l'access list en pensant
  - à l'ordre des conditions
  - que par défaut les paquets sont détruits
- appliquer l'access list sur l'interface concernée et dans la bonne direction

ACL étendue : au niveau 3 (ip)

access-list `num` {**deny** | **permit**} `protocole source [wildcard] destination [wildcard]`

exemples

`access-list 142 permit ip host 192.168.1.5 172.16.0.0 0.0.255.255`

autorise le trafic de l'hôte `192.168.1.5` vers le réseau `172.16.0.0/16`

`access-list 142 deny 192.168.1.0 0.0.0.255 172.16.0.0 0.0.255.255`

interdit le trafic du réseau `192.168.1.0/24` vers `172.16.0.0/16`

ACL étendue : au niveau 4 (udp/tcp)

access-list `num` {**deny**|**permit**} `protocole source [wildcard] [port_sce] destination [wildcard] [port_dest]`

exemples

`access-list 142 deny tcp any any`

interdit le trafic TCP

`access-list 142 permit udp any host 172.16.0.10 eq 53` autorise les paquets UDP vers l'hôte `172.16.0.10` pour le port destination 53 (DNS)

ACL étendue : principes d'utilisation

- principes similaire aux ACL standard mais
  - placer l'ACL étendue près de la source des paquets filtrés pour économiser de la bande passante

ACL nommée

ip access-list {standard | extended} `nom`

exemples

```
ip access-list standard john
permit 192.168.1.5
deny 192.168.0.0 0.0.0.255

ip access-list extended emma
permit host 192.168.1.5 172.16.0.0 0.0.255.255 eq 80
```

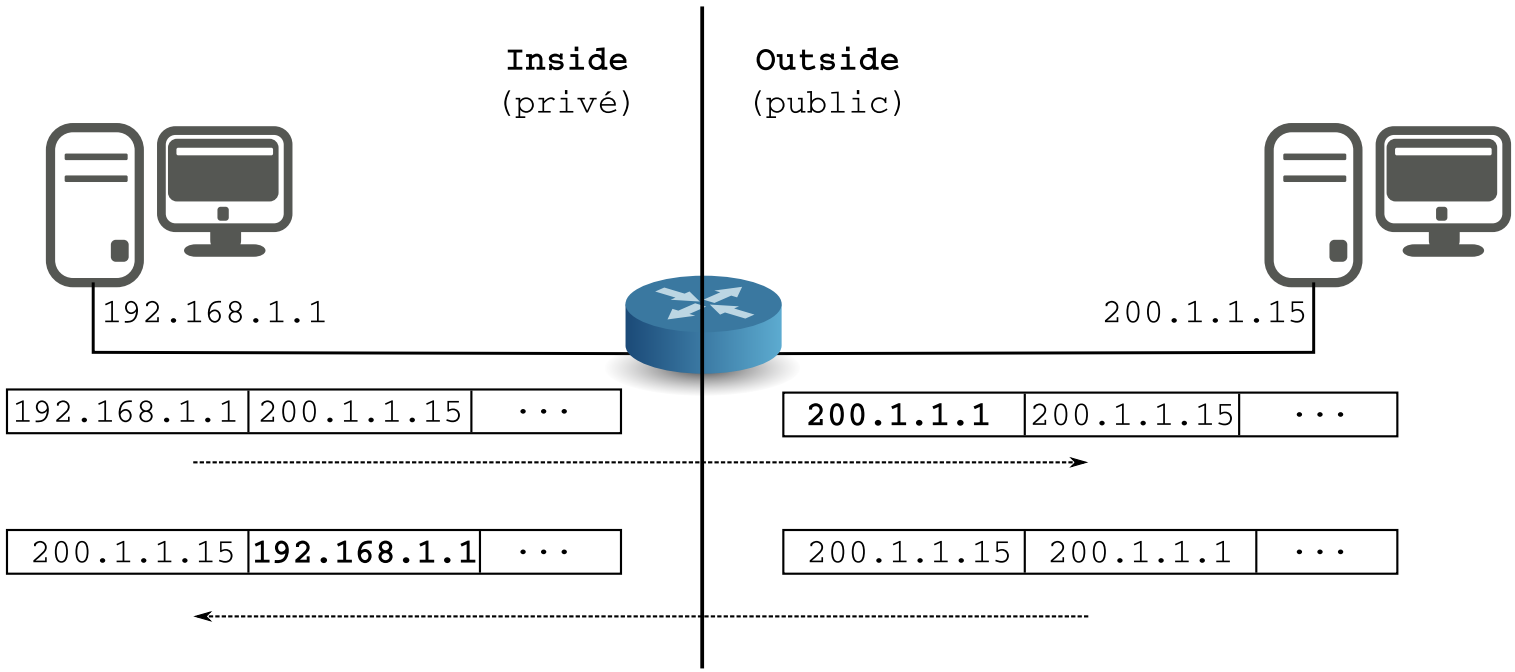
numéro de ligne

les lignes d'ACL sont automatiquement numérotées ce qui permet d'enlever une ligne spécifique ou d'en ajouter à l'endroit où l'on souhaite.

Motivation

- Permettre l'accès à Internet pour des hôtes qui n'ont pas d'adresse publique
- Masquer les adresses utilisées dans le réseau
- Solution de transition lors d'une renumérotation

Principe


Inside
(privé)
Outside
(public)
192.168.1.1
200.1.1.15
192.168.1.1 | 200.1.1.15 | ···
200.1.1.1 | 200.1.1.15 | ···
200.1.1.15 | 192.168.1.1 | ···
200.1.1.15 | 200.1.1.1 | ···

Types de translation

NAT statique

L'association adresse privée - adresse publique est fixe.

NAT dynamique

Le routeur pioche dans un lot d'adresses publiques disponibles.

NAPT - translation par port

Plusieurs adresses sont projetées sur la même adresse publique en faisant varier les numéros de port.

NAT statique

ip nat inside source static `@ip-privée @ip-publique`

exemple

```
ip nat inside source static 192.168.1.5 200.10.1.1
```

NAT : configuration des interfaces

ip nat { inside | outside }

Marquage des interfaces

Pour que le routeur connaisse la frontière entre la partie privée et publique, il faut marquer toutes les interfaces.

NAT dynamique

1. Définition du lot d'adresses publiques
   ip nat pool `nom @ip-début @ip-fin {`**netmask** `masque |`
   prefix-length `taille-prefixe}`
2. Définition d'une ACL qui indique les machines qui ont accès à la translation d'adresse
3. Définition de la règle de translation
   ip nat inside source list `num_acl` **pool** `nom`
4. marquer les interfaces

NAT par port - avec lot d'adresses

1. Définition du lot d'adresses publiques
   ip nat pool `nom @ip-début @ip-fin {`**netmask** `masque |`
   prefix-length `taille-prefixe}` **overload**
2. Définition d'une ACL qui indique les machines qui ont accès à la translation d'adresse
3. Définition de la règle de translation
   ip nat inside source list `num_acl` **pool** `nom`
4. marquer les interfaces

NAT par port - sur l'interface de sortie

1. Définition d'une ACL qui indique les machines qui ont accès à la translation d'adresse
2. Définition de la règle de translation
   ip nat inside source list `num_acl` **interface** `nom_interface` **overload**
3. marquer les interfaces

Contrôle de la translation

- effacer la table de translations (dynamique)
  clear ip nat translation *
- voir les translations en cours
  show ip nat translations
- voir les statistiques
  show ip nat statistics
- déboguer la translation
  debug ip nat