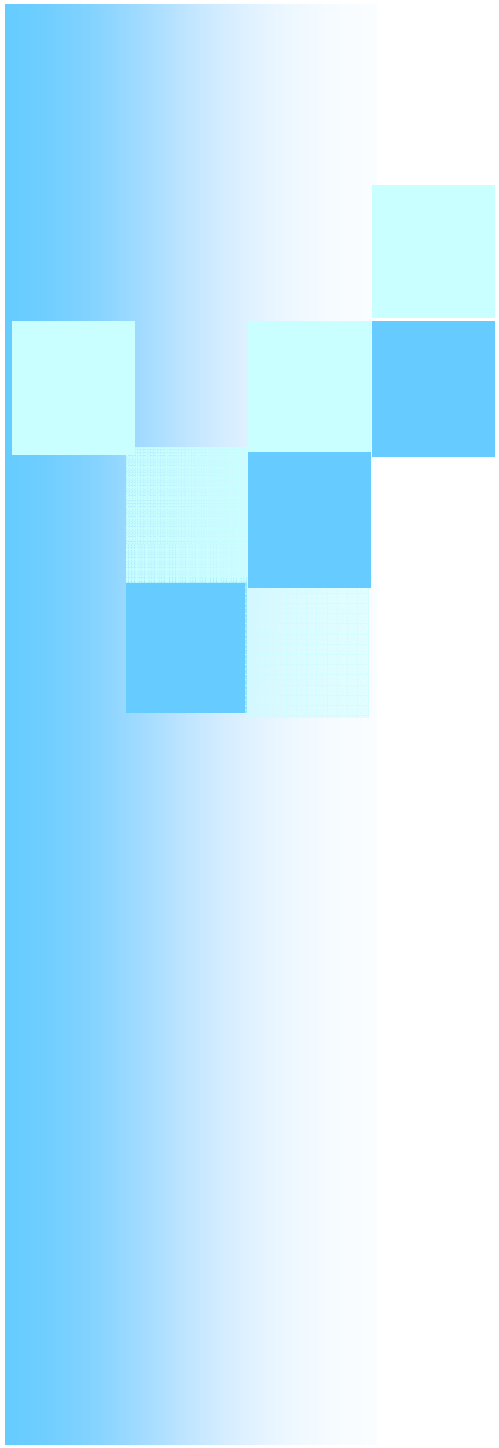
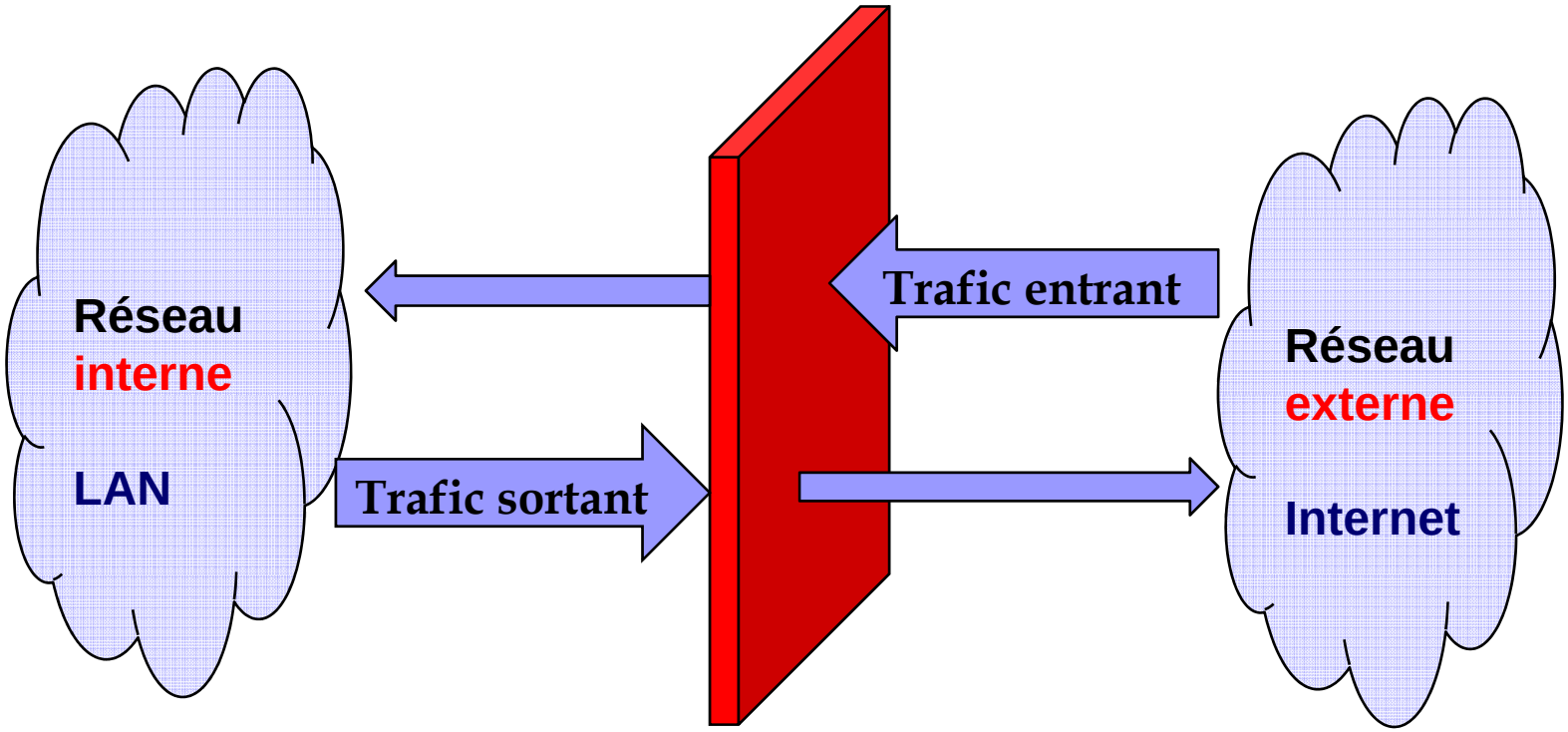


Chapitre 6

Contrôle d'accès
Filtrage et ACL

Filtrage de paquets: principe

- Filtrage du trafic entrant et du trafic sortant
  - Le firewall laisse passer certains paquets et rejette d'autres paquets selon sa politique de sécurité



Firewall
matériel ou logiciel

Filtrage de paquets: principe

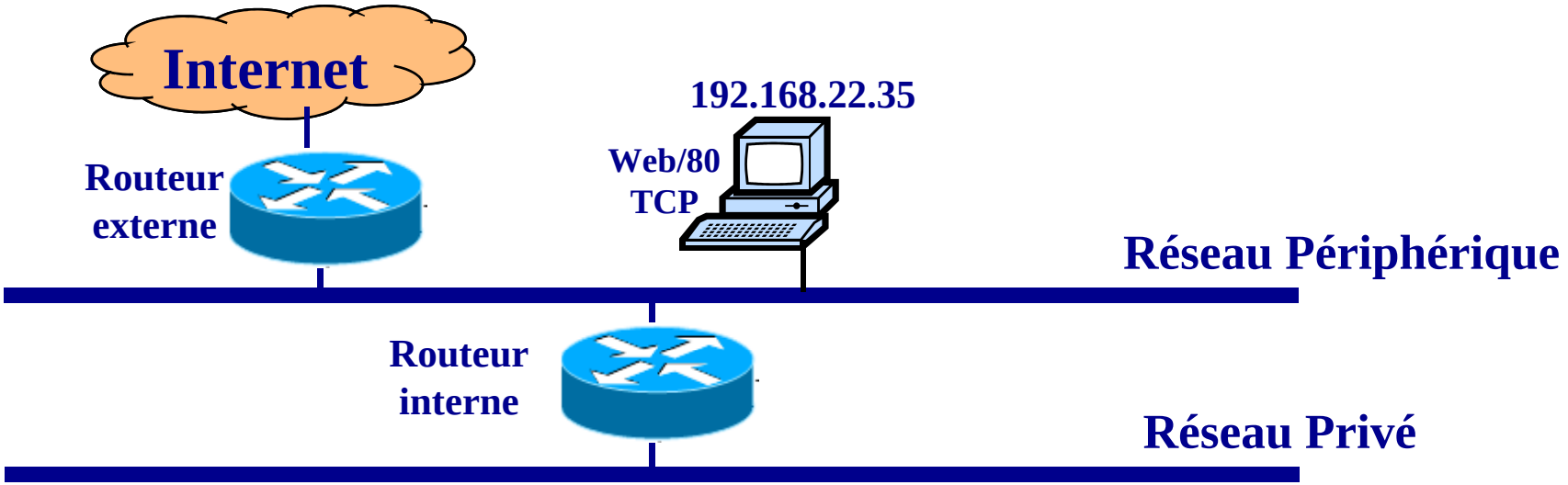
- Le filtrage se fait en analysant les en-têtes des protocoles, en priorité IP, UDP et TCP.

- En général, on définit une règle de filtrage en considérant
  1. **Adresse IP source**
  2. **Adresse IP destination**
  3. **Port source**
  4. **Port destination**
  5. **Protocole encapsulé (ICMP, UDP, TCP…)**
  6. **Flag ACK (de TCP)**
  7. **Type du message ICMP**

- A chaque règle de filtrage est associé une action:
  - Laisser passer le paquet ou
  - Détruire/Rejeter le paquet

Exemple de règles de filtrage

- Politique: Autoriser l'extérieur à accéder au service web sur le réseau périphérique



Règle	Direction paquet	IP Source	IP Dest	Prot	Port Source	Port Dest	ACK=1	Action
A	**Sortant**	**192.168.22.35**	**Toutes**	**TCP**	**80**	**> 1023**	**Oui**	**Autoriser**
B	**Entrant**	**Toutes**	**192.168.22.35**	**TCP**	**> 1023**	**80**	**---**	**Autoriser**
C	**Toutes**	**Toutes**	**Toutes**	**Tous**	**Tous**	**Tous**	**---**	**Refuser**

Types de filtrage

- Filtrage sans état: Stateless
  - ➔ Filtrage simple: Regarder chaque paquet à part et le comparer à une liste de règles préconfigurées (ACL)
  - ➔ Implémenté sur les routeurs et les systèmes d'exploitations
  - Limites
    - ➔ Utiliser un trop grand nombre de règles pour que le Firewall offre une réelle protection
    - ➔ Sensibles aux attaques IP spoofing / IP flooding; attaques DoS

Types de filtrage

- Filtrage à état: Statefull
  - ➔ Tracer les sessions et les connexions dans des tables d'états internes au Firewall
  - ➔ Décider en fonction des états de connexions
  - ➔ Exemple: vérifier que chaque paquet d'une connexion est bien la suite du précédent paquet et la réponse à un paquet dans l'autre sens
  - ➔ L'application des règles est possible sans lire les ACL à chaque fois (les paquets d'une connexion actives seront acceptés)

Types de filtrage

- Filtrage applicatif (firewall de type proxy)
  - ➔ Réalisé au niveau de la couche Application
  - ➔ Permet d'extraire les données du protocole applicatif pour les étudier
  - ➔ Chaque protocole est filtré par un processus dédié
  - Limites
    - Problèmes de performance pour les réseaux à grand trafic

Processus de développement de filtres

Processus de développement de filtres

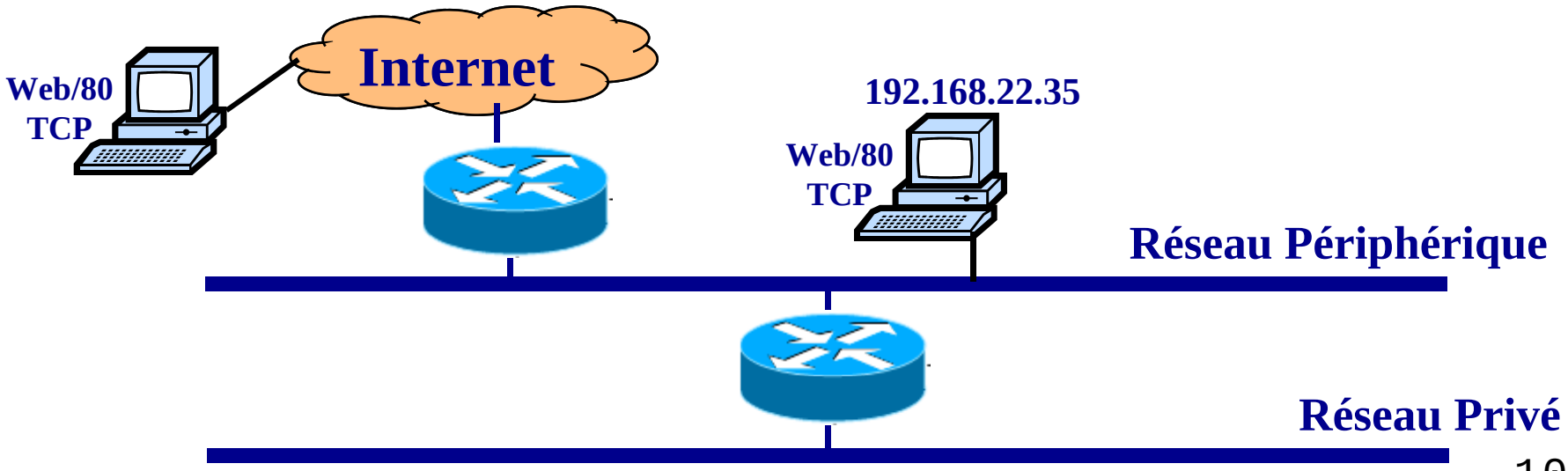
- Définition des règles de filtrage
  - Utiliser le maximum de critères (@IP, port, ACK…etc)
  - → Permet de mieux lutter contre les attaques
- Pour chaque service interne et externe
  - Définir des règles pour autoriser les utilisateurs interne à accéder à des services externes
  - Définir des règles pour autoriser des utilisateurs externes à accéder à des serveurs (services) sur le réseau interne
- Pour un service à autoriser
  - Accepter le flux dans les deux sens (client→serveur et serveur→client)
- Pour un service à bloquer
  - Il suffit de bloquer le flux du client→serveur

Processus de développement de filtres

- Exemple:
  - Soit la politique:

    Accepter HTTP en entrée et en sortie et rien d'autre.

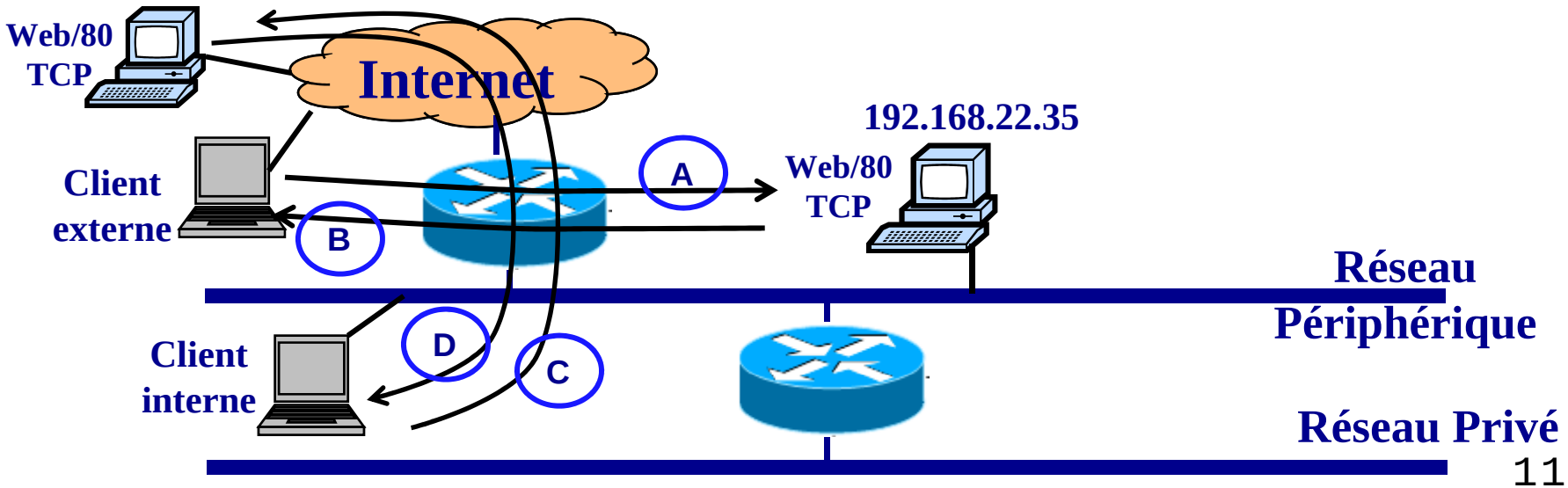
    - Autoriser les utilisateurs internes à accéder aux serveurs web externes
    - Autoriser les utilisateurs externes à accéder au serveur web interne
  - Objectif : développer les règles correspondantes



Processus de développement de filtres

- Exemple de règles

Règle	Direction	@ source	@ Dest.	Protocole	Port dest.	Action
A	Entrant	Externe	192.168.22.35	TCP	80	Autoriser
B	Sortant	192.168.22.35	Externe	TCP	>1023	Autoriser
C	Sortant	Interne	Externe	TCP	80	Autoriser
D	Entrant	Externe	Interne	TCP	>1023	Autoriser
E	Toutes	Toutes	Toutes	Tous	Tous	Refuser

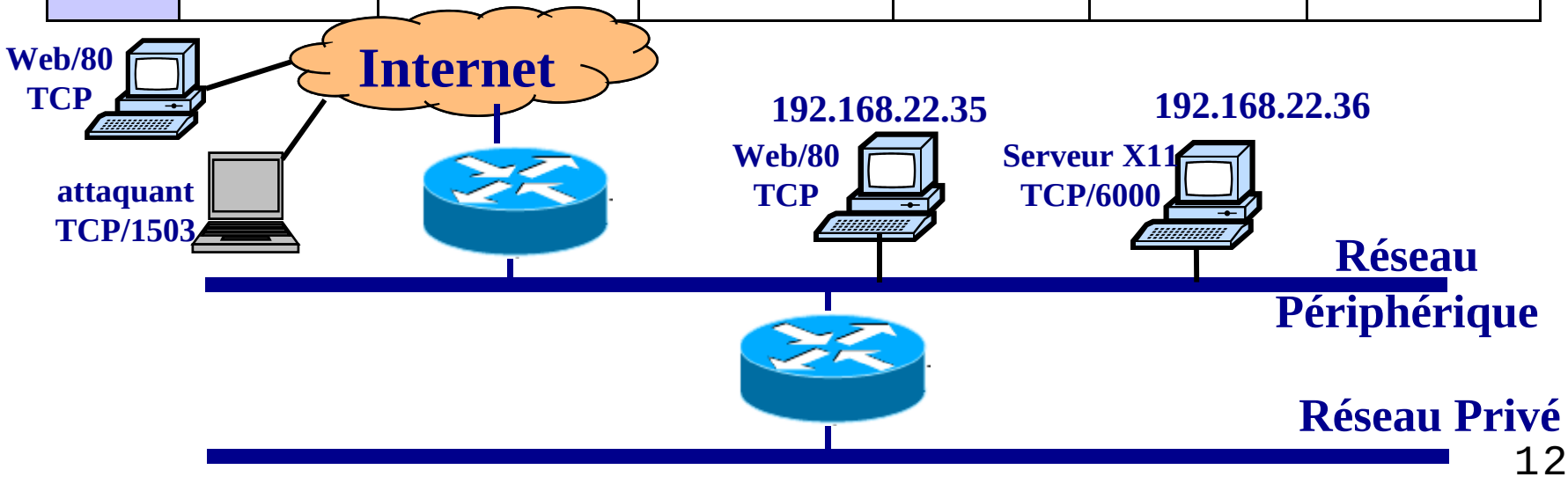


Processus de développement de filtres

Question

- Ces règles autorisent-elles les connexions dont les ports source et destination sont supérieurs à 1023? (ce qui n'était pas prévu)

Règle	Direction	@ source	@ Dest.	Protocole	Port dest.	Action
A	Entrant	Externe	192.168.22.35	TCP	80	Autoriser
B	Sortant	192.168.22.35	Externe	TCP	>1023	Autoriser
C	Sortant	Interne	Externe	TCP	80	Autoriser
D	Entrant	Externe	Interne	TCP	>1023	Autoriser
E	Toutes	Toutes	Toutes	Tous	Tous	Refuser



Processus de développement de filtres

- Une solution
  - Examiner aussi le port source

Règle	Direction	@ source	@ Dest.	Protocole	Port src.	Port dest.	Action
A	Entrant	Externe	192.168.22.35	TCP	>1023	80	Autoriser
B	Sortant	192.168.22.35	Externe	TCP	80	>1023	Autoriser
C	Sortant	Interne	Externe	TCP	>1023	80	Autoriser
D	Entrant	Externe	Interne	TCP	80	>1023	Autoriser
E	Toutes	Toutes	Toutes	Tous	Tous	Tous	Refuser

- Mais
  - Un attaquant peut utiliser le port 80 comme port source client puis se connecte au serveur X11/ port 6000

  ➔ Il réussira (règle D et C)

Processus de développement de filtres

- Raffinement de la solution
  - Examiner aussi le flag ACK (ACK=0 seulement dans le premier paquet envoyé du **client (port>1023)** vers le serveur)

Règle	Direction	@ source	@ Dest.	Protocole	Port src.	Port dest.	ACK=1	Action
A	Entrant	Externe	192.168.22.35	TCP	>1023	80	---	Autoriser
B	Sortant	192.168.22.35	Externe	TCP	80	>1023	oui	Autoriser
C	Sortant	Interne	Externe	TCP	>1023	80	---	Autoriser
D	Entrant	Externe	Interne	TCP	80	>1023	oui	Autoriser
E	Toutes	Toutes	Toutes	Tous	Tous	Tous	---	Refuser

- Mais
  - Un attaquant peut utiliser le port 80 comme port source client puis se connecte au serveur X11/ port 6000 en fixant **ACK à 1**

➔ Réussira t-il à se connecter au serveur (considérer les règles D et C)

Processus de développement de filtres

- Réponse:
  - Le paquet passera au travers les filtres,

  Mais,

  - La destination pensera que le paquet appartient à une connexion existante.

  - Quand la destination essayera de faire correspondre le paquet avec une connexion existante, elle échouera et le **paquet sera ignoré**

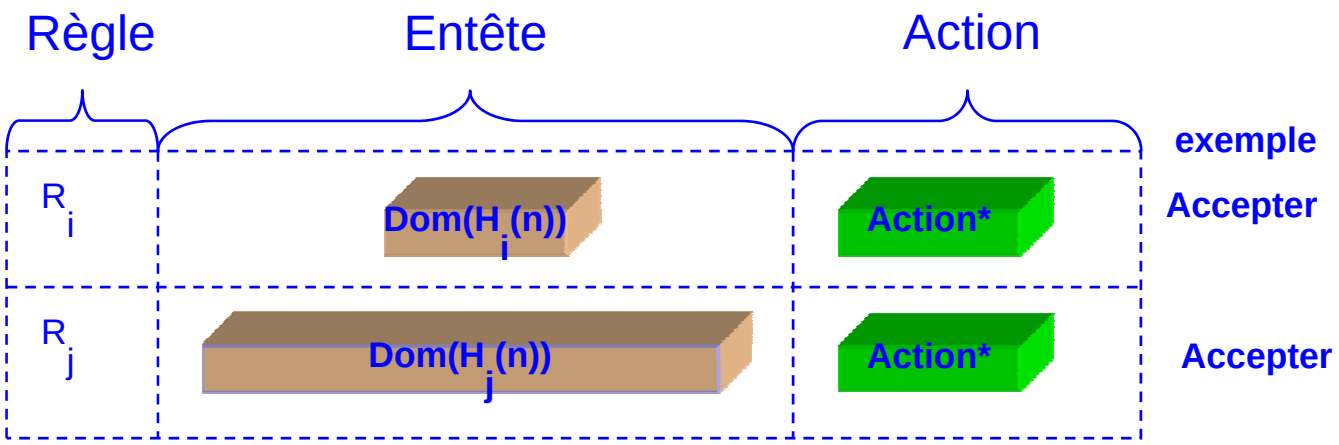
Processus de développement de filtres

- Conclusion:
  - Il faut considérer les règles de filtrage comme un seul bloc
  - Il faut utiliser le maximum de critères de filtrage
  - Le flag ACK est important pour les connexions TCP

Détection et corrections des anomalies dans les règles de filtrage

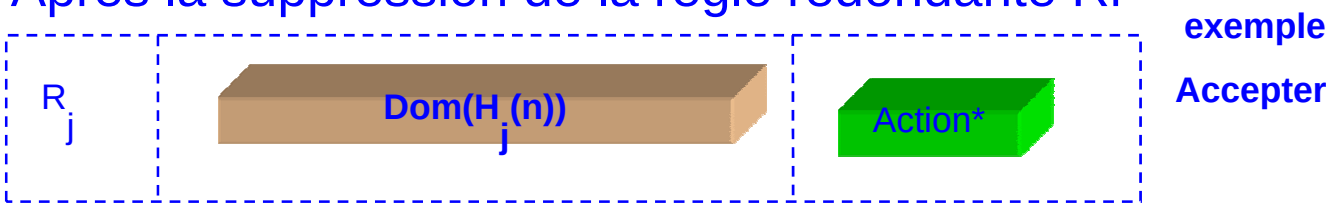
- Anomalie redondance
- Anomalie Masquage
- Anomalie Généralisation
- Anomalie Corrélation

Anomalie redondance

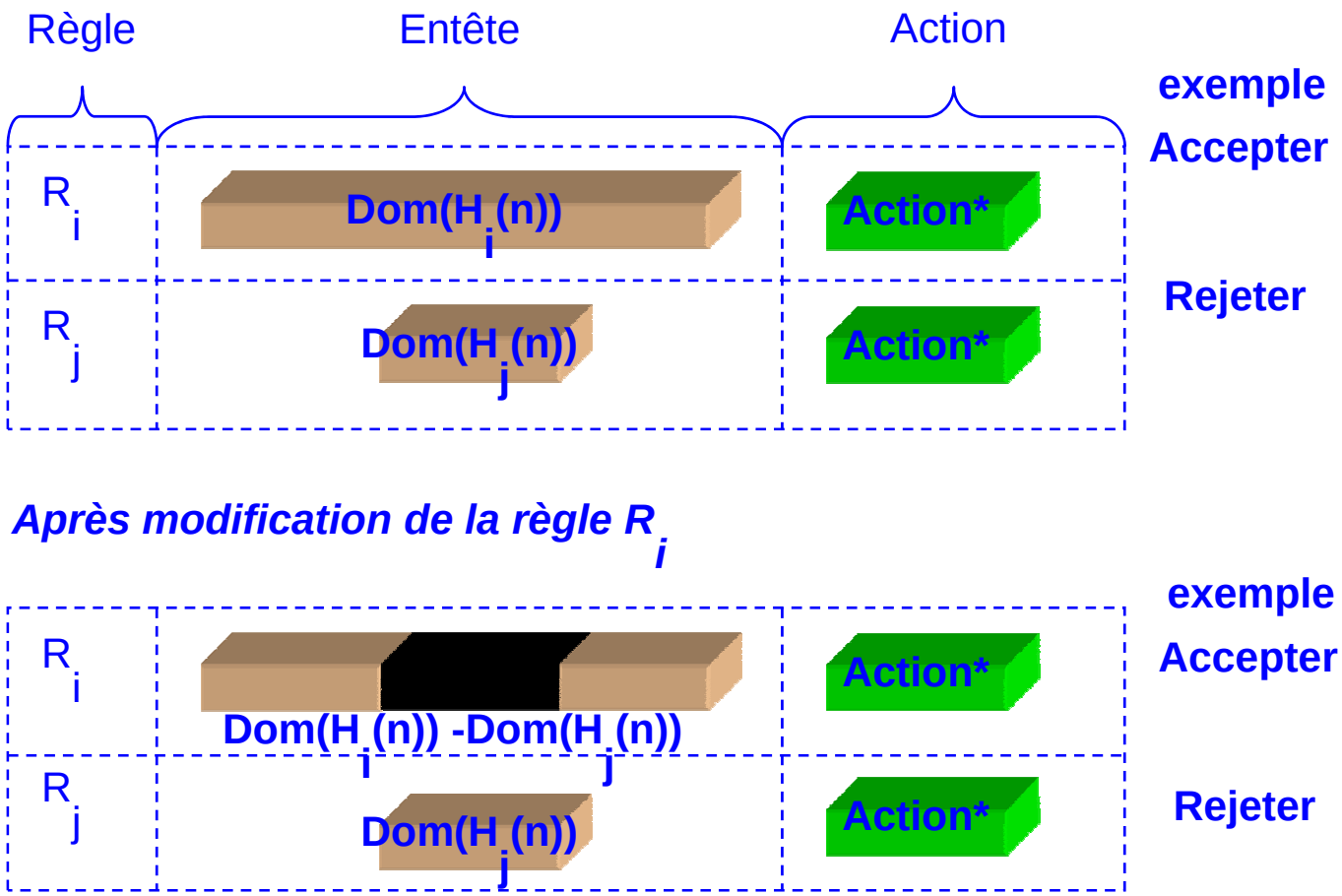


* : Les règles Ri et Rj ont la **même action (Accepter ou Rejeter)**

Après la suppression de la règle redondante Ri

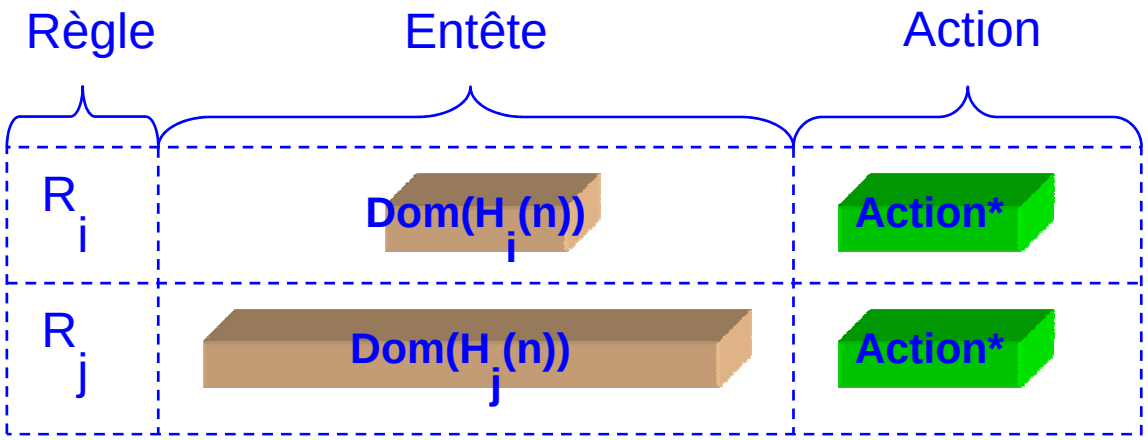


Anomalie masquage



*** : Les règles Ri et Rj ont des actions différentes**

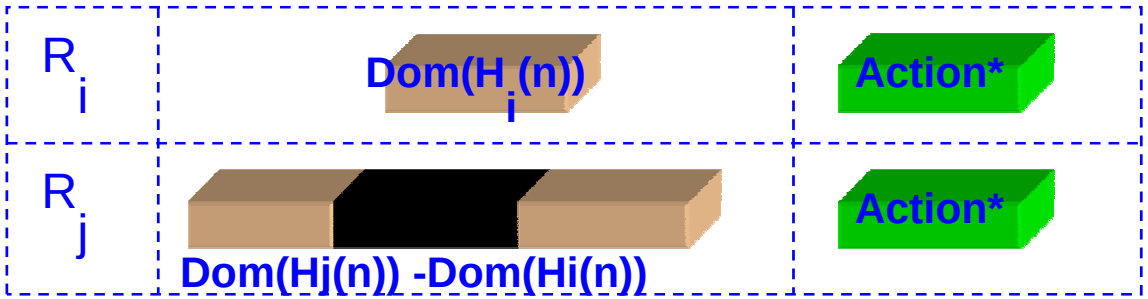
Anomalie généralisation



exemple
Accepter

Rejeter

Après modification de la règle R j

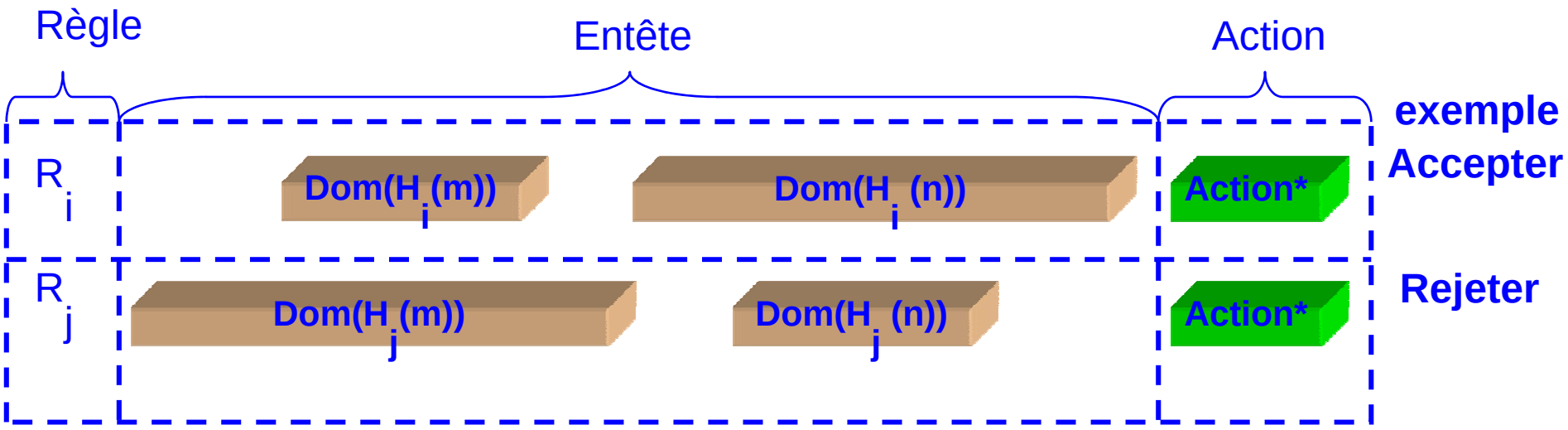


exemple
Accepter

Rejeter

*** : Les règles Ri et Rj ont des actions différentes**

Anomalie corrélation

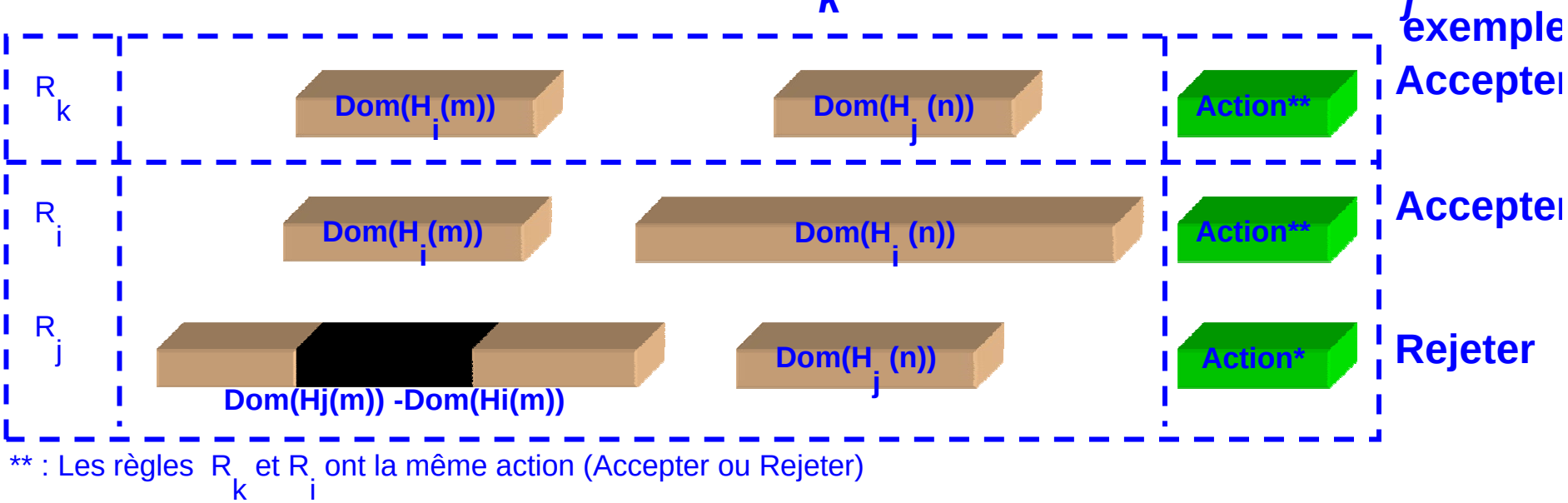


* : Les règles R_i et R_j ont des **actions différentes**

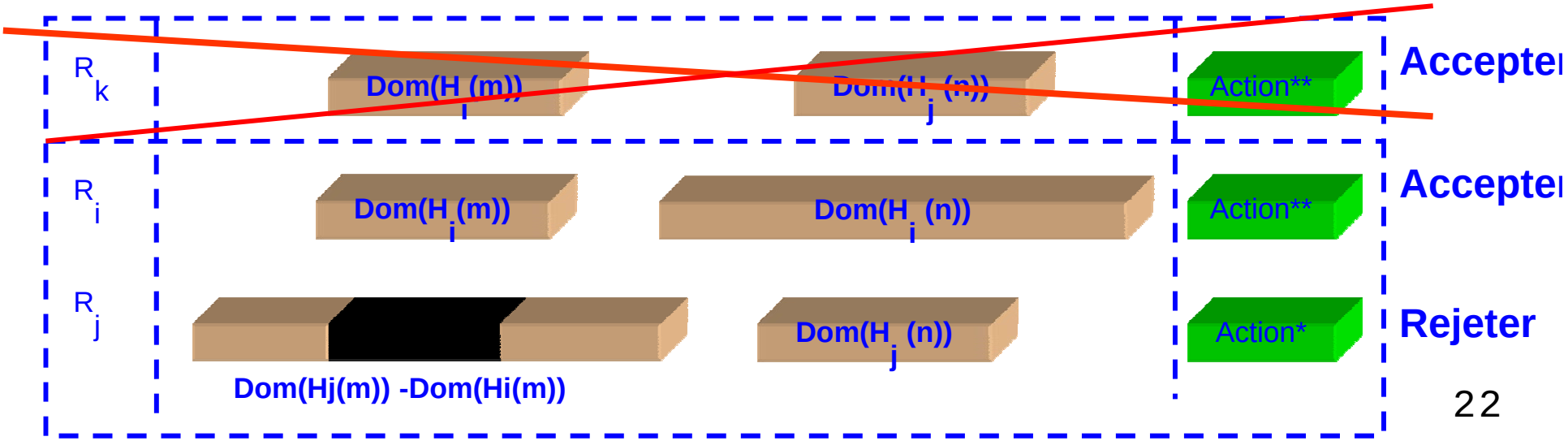
Note : m et n sont des champs des entêtes H_i et H_j respectivement

Anomalie corrélation

Cas 1 : création d'une nouvelle règles R_k et modification de la règle R_j

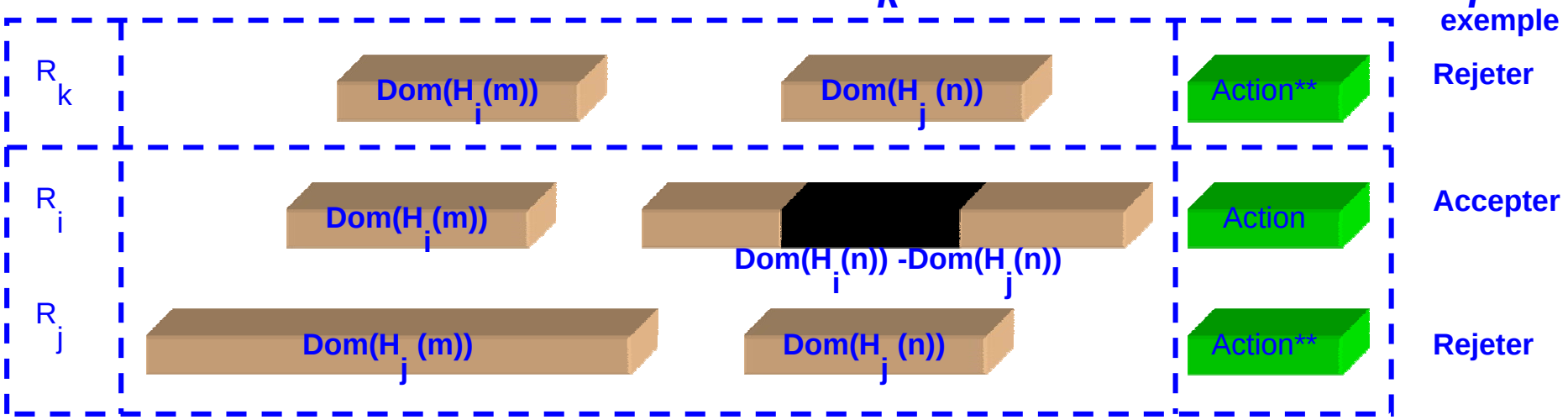


** : Les règles R_k et R_i ont la même action (Accepter ou Rejeter)

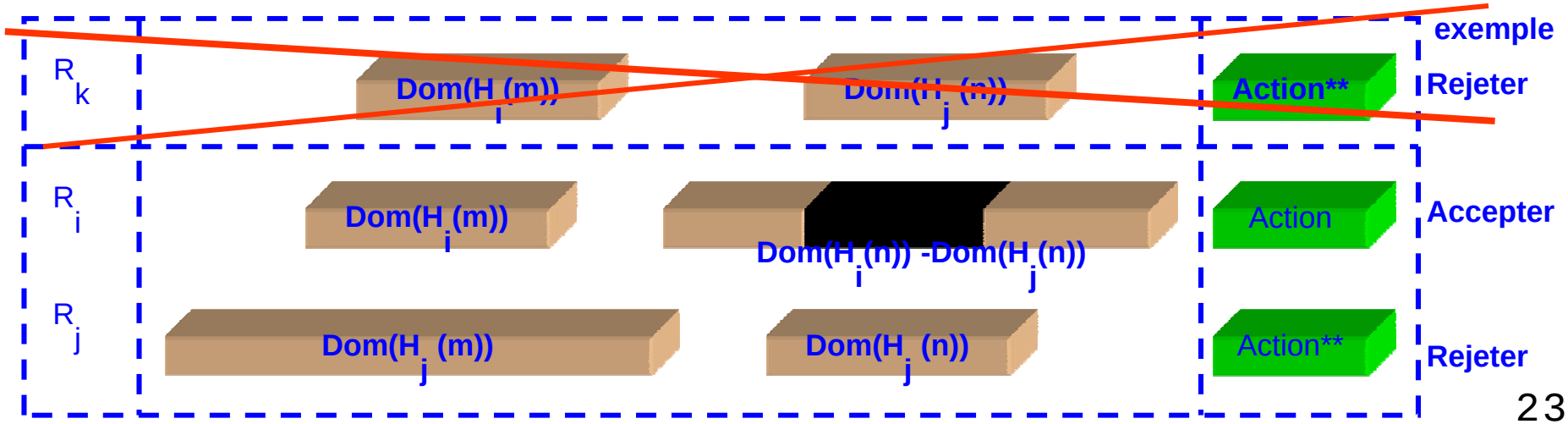


Anomalie corrélation

Cas 2 : création d'une nouvelle règles R_k et modification de la règle R_i



** : Les règles R_k et R_j ont la **même action (Accepter ou Rejeter)**



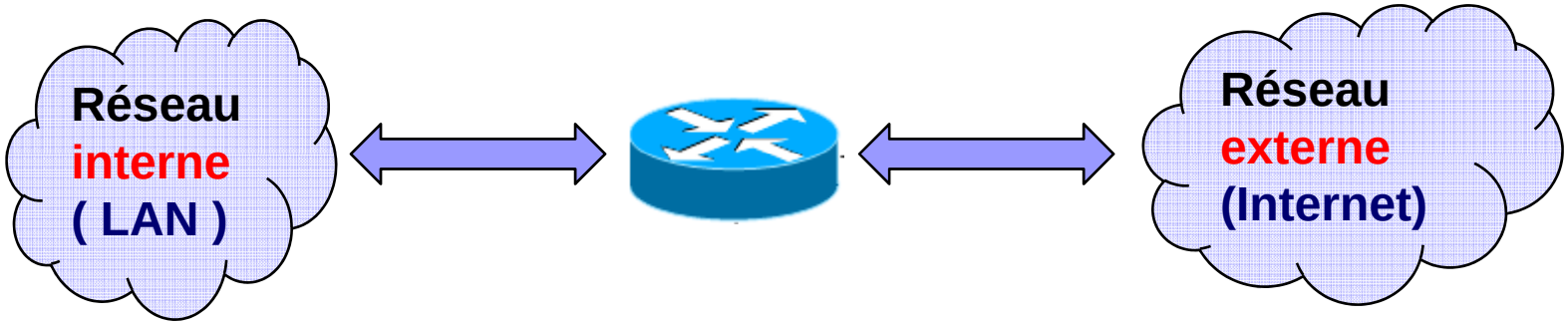
Firewalls matériels / logiciels

Firewall matériel / logiciels

- Firewalls matériels
  - Routeurs filtrants
  - Firewalls sous forme de boîtiers

- Firewall logiciels
  - Firewall professionnels
    - Firewall libre : Netfilter / iptables
    - Firewall commercial : CheckPoint Firewall-1, ASA, PIX
  - Firewall personnels
    - Kerio, Zone Alarm…

Firewall matériel: Routeurs filtrants



- Un routeur filtrant
  - Examine chaque paquet afin de déterminer s'il doit l'acheminer ou l'abandonner
  - Bien adapté aux PME
  - Pas de fichiers logs et pas de statistiques

- La fonction de filtrage est implémentée dans la plupart des routeurs du marché
  - Sous forme de listes d'accès ACL
  - En utilisant une syntaxe spécifique par routeur

Firewall matériel: Routeurs filtrants

- Inconvénients
  - Accès à des parties limitées des entêtes des paquets.
  - Aucune information de l ’état d ’une communication de bout en bout.
  - « IP-Spoofing Ready »: pas d’authentification de l’origine du paquet: ne sait pas si l ’auteur du paquet est bien celui qui l’émet
  - Sensibles aux attaques par fragmentation

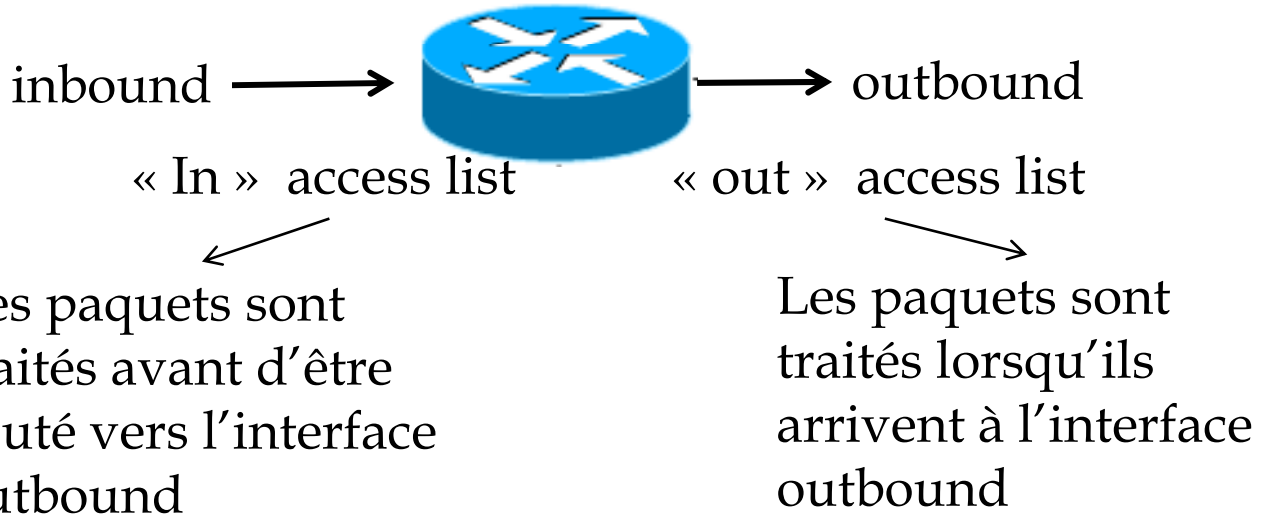
Firewall matériel: Firewall sous forme boîtiers

- Conçus uniquement pour faire du filtrage
- OS spécifique, associé au boîtier
- Rapidité de traitement
- Supportent rarement les interfaces WAN → nécessité d'être associés à des routeurs pour la connectivité
- Exemple :
  - Cisco ASA (Adaptive Security Appliance)
  - Cisco PIX (Private Internet eXchange)

Access Control Lists (ACL)

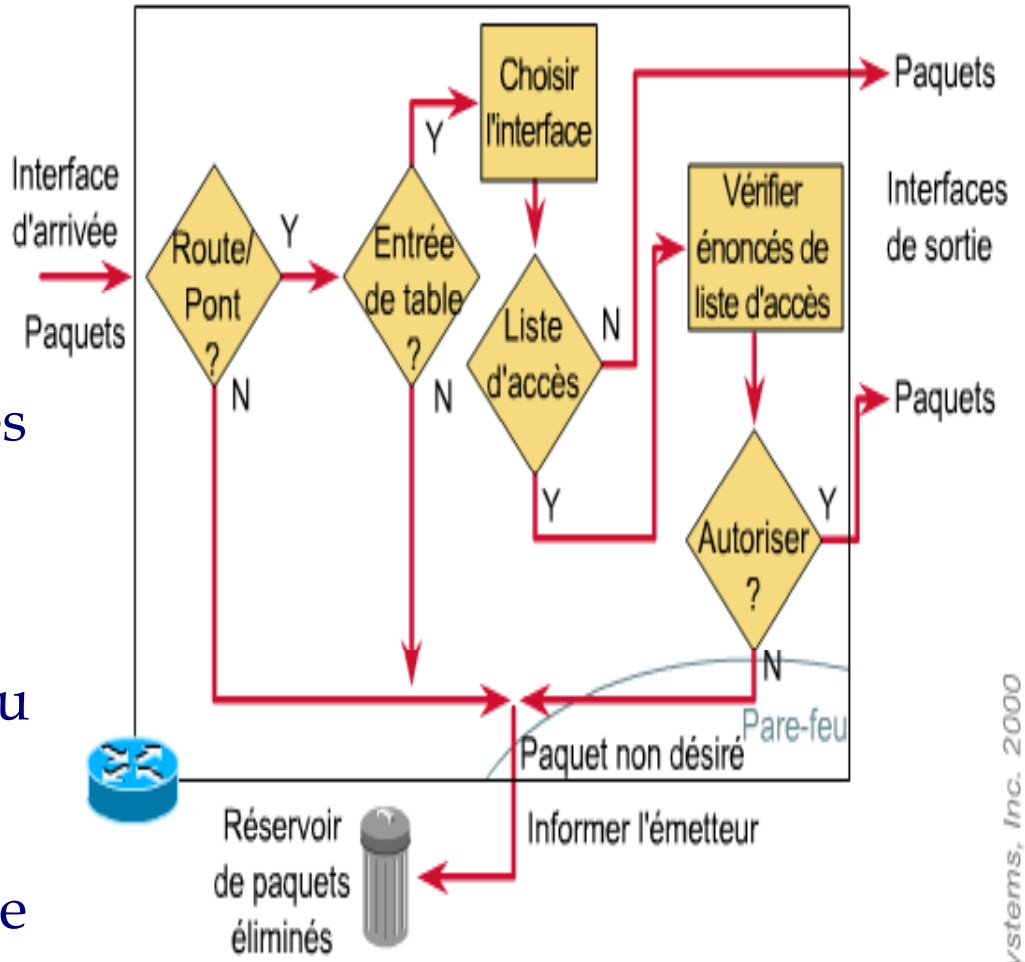
ACL dans les routeurs

- Permet d'implémenter des règles de filtrage dans les routeurs → Contrôler l'accès entre différents réseaux
- Une ACL créée doit être associée à une interface du routeur où le filtrage sera exécuté.
  - Interface **in** (incoming: paquets entrant dans le routeur)
  - Interface **out** (outcoming: paquets sortant du routeur)



ACL: Processus de contrôle des paquets

- Un paquet est comparé aux règles de l'ACL d'une manière séquentielle Top-Down
- La comparaison s'arrête dès qu'un paquet vérifie l'une des règles de l'ACL
- L'action (permit/deny) de la règle trouvée est appliquée au paquet
- Les ACL se terminent par une règle « deny all » implicite pour rejeter les paquets qui ne vérifient aucune règle



ACL numbers

```
Router(config)#access-list ?
```

```
  <1-99>      IP standard access list
  <100-199>   IP extended access list
  <200-299>   Protocol type-code access list
  <300-399>   DECnet access list
  <400-499>   XNS standard access list
  <500-599>   XNS extended access list
  <600-699>   Appletalk access list
  <700-799>   48-bit MAC address access list
  <800-899>   IPX standard access list
  <900-999>   IPX extended access list
  <1000-1099> IPX SAP access list
  <1100-1199> Extended 48-bit MAC address access list
  <1200-1299> IPX summary address access list
```

Standard IP access lists (1-99)

- Filtrage en se basant sur l'adresse IP source uniquement
- Se placent près de la destination
- Syntaxe

  - Créer la liste d'accès

Router(config)# access-list numéro-liste-d'accès {deny|permit} source [wildcard mask] [log]

  - Associer la liste d'accès à une interface du routeur:

Router(config)# interface [port du routeur]

Router(config-if)# ip access-group numéro-liste-d'accès {in/out}

Standard IP access lists (1-99)

- Syntaxe

  Router(config)# access-list numéro-liste-d'accès {deny|permit} source [wildcard mask] [log]

  - Source:

  Hostname or A.B.C.D; any (n'importe quel hôte), host (hôte particulier)

  - wildcard mask (32 bits)
    - Les bits '0' signifient que les mêmes positions de bits doivent être vérifiées (match)
    - Les bits '1' signifient que les bits de mêmes positions sont ignorés
    - Exemples

Router(config)# access-list 14 deny 192.168.16.0 0.0.0.255 (tous les hôtes)

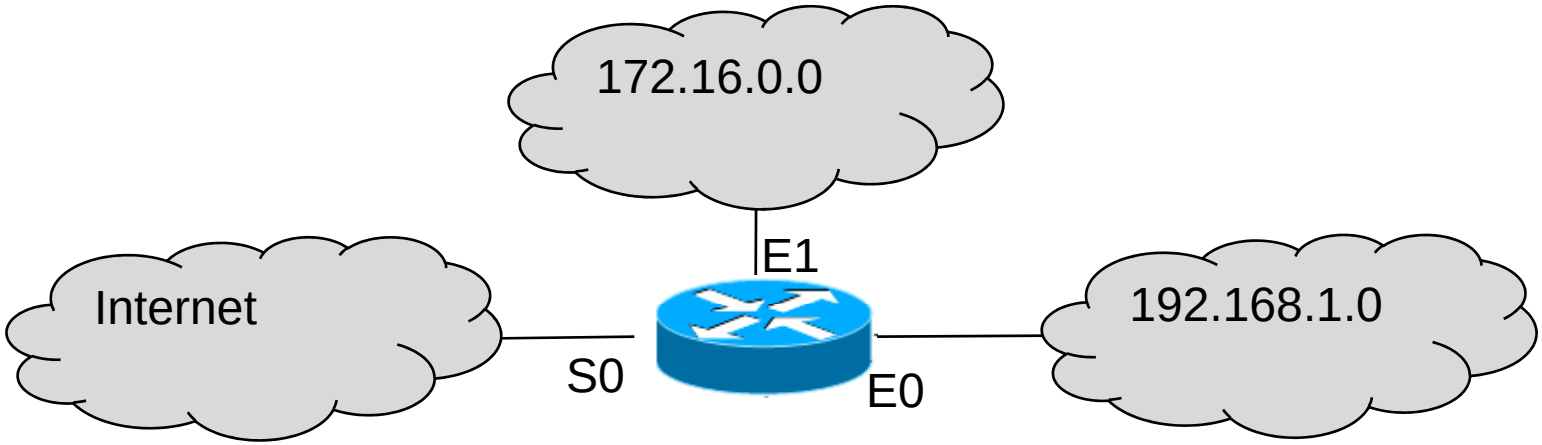
Router(config)# access-list 14 deny 192.168.16.0 0.0.0.127 (1ère moitié)

Router(config)# access-list 14 deny 192.168.16.128 0.0.0.127 (2ème moitié)

Standard IP access lists (1-99)

- Exemple:
  - Permettre l'acheminement du trafic du réseau 192.168.1.0 (vers Internet et vers 172.16.0.0)

```
Router(config)# access-list 11 permit 192.168.1.0 0.0.0.255
Router(config)# int S0
Router(config-if)# ip access-group 11 out
Router(config)# int E1
Router(config-if)# ip access-group 11 out
```



Extended IP access lists

- Filtrage en se basant sur:
  - @IP source et @IP destination
  - Port source et port destination (filtrage par service)
  - Type de protocole de transport (TCP, UDP)
- Se placent près de la source
- Syntaxe
  - Créer la liste d'accès

Router(config)# access-list numéro-liste-d'accès {deny|permit} protocol source [source mask] destination [destination mask] [operator operand]

  - Associer la liste d'accès à une interface du routeur:

Router(config)# interface [port du routeur]

Router(config-if)# ip access-group numéro-liste-d'accès {in/out}

Extended IP access lists (syntaxe)

Router(config)# access-list numéro-liste-d'accès {deny|permit} *protocol* source [source mask] destination [destination mask] [operator operand]

Router(config)#access-list 112 deny ?

<0-255>	An IP protocol number
eigrp	Cisco's EIGRP routing protocol
gre	Cisco's GRE tunneling
icmp	Internet Control Message Protocol
igmp	Internet Gateway Message Protocol
igrp	Cisco's IGRP routing protocol
ip	Any Internet Protocol
ipinip	IP in IP tunneling
nos	KA9Q NOS compatible IP over IP tunneling
ospf	OSPF routing protocol
tcp	Transmission Control Protocol
udp	User Datagram Protocol

Extended IP access lists (syntaxe)

Router(config)# access-list numéro-liste-d'accès {deny|permit} protocol *source* [source mask] *destination* [destination mask] [operator operand]

Router(config)#access-list 112 deny tcp ?

A.B.C.D	Source address
any	Any source host
host	A single source host

Router(config)#access-list 112 deny tcp any ?

A.B.C.D	Destination address
any	Any destination host
host	A single destination host

Extended IP access lists (syntaxe)

Router(config)# access-list numéro-liste-d'accès {deny|permit} protocol source [source mask] destination [destination mask] [*operator* operand]

Router(config)#access-list 112 deny tcp any host 172.16.30.2 ?

eq	Match only packets on a given port number
established	Match established connections
fragments	Check fragments
gt	Match only packets with a greater port number
log	Log matches against this entry
log-input	Log matches against this entry, including input interface
lt	Match only packets with a lower port number
neq	Match only packets not on a given port number
precedence	Match packets with given precedence value
range	Match only packets in the range of port numbers
tos	Match packets with given TOS value

Extended IP access lists (syntaxe)

Router(config)# access-list numéro-liste-d'accès {deny|permit} protocol source [source mask] destination [destination mask] [operator *operand*]

Router(config)#access-list 112 deny tcp any host 172.16.30.2 eq ?

daytime	Daytime (13)
domain	Domain Name Service (53)
echo	Echo (7)
ftp File	Transfer Protocol (21)
irc	Internet Relay Chat (194)
lpd	Printer service (515)
smtp	Simple Mail Transport Protocol (25)
sunrpc	Sun Remote Procedure Call (111)
telnet	Telnet (23)
www	World Wide Web HTTP,80)

…….etc

Extended IP access lists

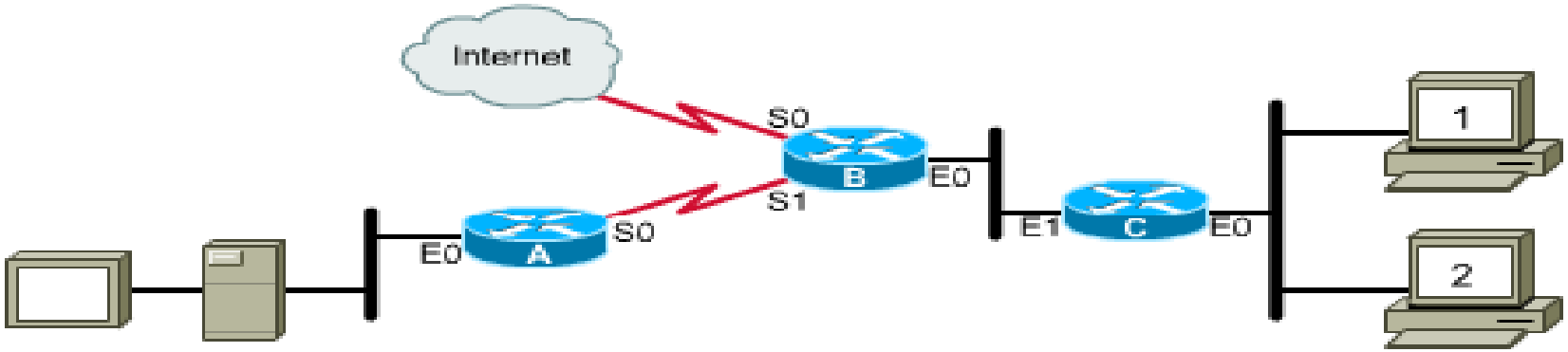
- Exemple 1: refuser l'accès du réseau 221.23.123.0 au serveur FTP (TCP/21) 198.150.13.34
  - → Extended ACL→ Placer la règle près de la source
  - → Ecrire l'ACL dans le routeur C et l'appliquer à l'interface E0

Router(config)#access-list 113 deny tcp 221. 23.123.0 0.0.0.255 host 198.150.13.34 eq 21

Router(config)#access-list 113 permit ip 221. 23.123.0 0.0.0.255 0.0.0.0 255.255.255.255

Router(config)# int E0

Router(config-if)# ip access-group 113 in



Server
IP: 198.150.13.34

Router A
E0: 198.150.13.56
S0: 223.11.34.123

Router B
E0: 222.21.38.100
S0: 220.100.5.122
S1: 223.11.34.124

Router C
E0: 221.23.123.96
E1: 222.21.38.123

Workstation 1
IP: 221.23.123.45

Workstation 2
IP: 221.23.123.40

Nommage des ACL

- Assigner des noms aux ACL

- Utile lorsqu'on a besoin de plus de 99 ACL

- Exemple

Router(config)#ip access-list standard nom_liste

Router(config-std-nacl)# deny host 172.16.2.3

→ Les paramètres access-list et access-list-number sont implicites

Router(config)# int E0

Router(config-if)# ip access-group nom_liste out

Vérification des ACLs

- La commande Show:
  - show access-lists
    - Montre toutes les ACLs configuré dans le routeur
  - show access-lists {name | number}
    - Montre l'ACL spécifié
  - show ip interface
    - Montre l'ACL appliqué à l'interface (inbound et outbound).
  - show running-config
    - Montre toutes les ACLs et les interfaces où elle sont appliquées

A retenir

- Assigner une seule ACL par interface, par protocole et par direction (une seule ACL inbound et une seule ACL outbound par interface)
- L'ajout de nouvelle lignes se fait à la fin de la liste
- Une ACL se termine par un deny any implicite → une liste d'accès doit contenir au minimum une ligne permit
- Les ACL ne permettent pas de filtrer le trafic généré par le routeur
- Placer les ACL standards près de la destination
- Placer les ACL eétendues près de la source