

S22 - Architecture des réseaux

Routage

Cédric Wemmert

IUT Robert Schuman – Département Informatique

wemmert@unistra.fr

2022

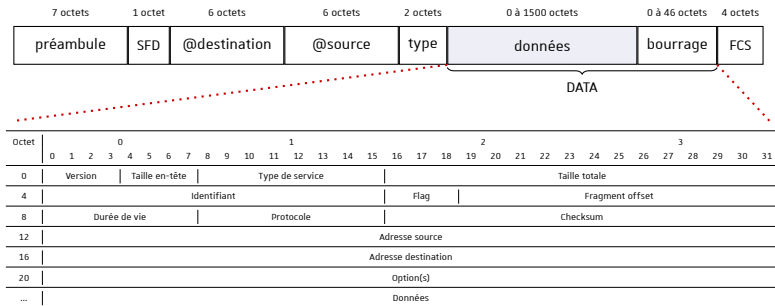
Le paquet IP

Octet	0				1										2										3							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Version			Taille en-tête			Type de service										Taille totale															
4	Identifiant															Flag			Fragment offset													
8	Durée de vie							Protocole										Checksum														
12	Adresse source																															
16	Adresse destination																															
20	Option(s)																															
...	Données																															

- Version : 4 ou 6
- Taille en-tête : exprimée en nombre de mots de 32 bits
- Identifiant, flag et fragment offset : cf. fragmentation des paquets IP
- Durée de vie (TTL) : décrétementée à chaque passage dans un routeur ; si TTL == 0, le paquet est rejeté (permet d'éviter les bouclages infinis)
- Protocole : identifiant du protocole dans les données
- Checksum : cf. M31...

Encapsulation

Le paquet IP est encapsulé dans une trame Ethernet :



Encapsulation

Par la suite nous utiliserons une version simplifiée des trames Ethernet :

Trame simplifiée

Couche Liaison	@MAC dest	MAC XX
	@MAC src	MAC YY
Couche Réseau	@IP src	XXX.XXX.XXX.XXX
	@IP dest	XXX.XXX.XXX.XXX
Couche Transport à Application	DATA	

ICMP

Internet Control and error Message Protocol

- protocole important et nécessaire d'IP
- responsable de deux fonctions principales :
 - tester l'accessibilité d'un équipement
 - rendre compte d'un problème réseau

Messages ICMP

- messages d'interrogation ou d'information : différents messages (demande d'écho, demande d'heure, etc.)
- messages d'erreur : transmis à la source de l'envoi générant l'erreur (destinataire inaccessible, temps dépassé, en-tête erronée, etc.)

ICMP

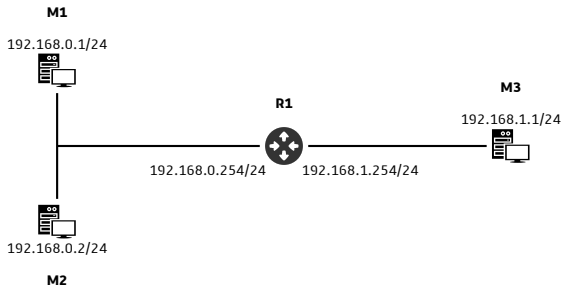
Exemple : la commande ping

Utilisation de messages ECHO REQUEST et ECHO REPLY

```
$ ping -c 2 130.79.80.40
PING sterne.iutrs.unistra.fr (130.79.80.40): 56 data bytes
64 bytes from 130.79.80.40: icmp_seq=0 ttl=54 time=44.400 ms
64 bytes from 130.79.80.40: icmp_seq=1 ttl=54 time=44.746 ms

--- 130.79.80.40 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 44.400/44.573/44.746/0.173 ms
```

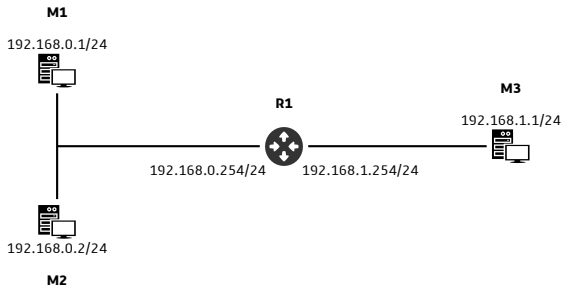
Scénarios



2 scénarios

- 1 M1 envoie un message à M2
- 2 M1 envoie un message à M3

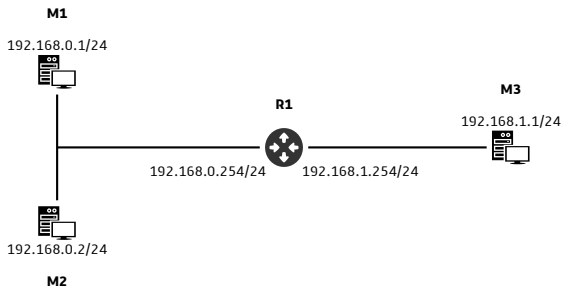
Configuration de l'interface



Configuration de l'interface de M1

- Adresse IP 192.168. 0. 1
- Masque de sous-réseau 255.255.255. 0

Scénario 1 : M1 envoie un message à M2



M1 détecte le sous-réseau

- M1 applique le masque de sous-réseau :
 - M1 : 192.168.0.1 ET 255.255.255.0 = 192.168.0.0
 - M2 : 192.168.0.2 ET 255.255.255.0 = 192.168.0.0

↪ M1 détecte que les deux machines sont dans le **même sous-réseau**

Scénario 1 : M1 envoie un message à M2

M1 et M2 dans le même sous-réseau : envoi d'une trame directement à M2

Trame simplifiée

Couche Liaison	@MAC dest	?
	@MAC src	?
Couche Réseau	@IP src	?
	@IP dest	?
Couches Transport à Application	DATA	

Scénario 1 : M1 envoie un message à M2

M1 et M2 dans le même sous-réseau : envoi d'une trame directement à M2

Trame simplifiée

Couche Liaison	@MAC dest	?
	@MAC src	MAC M1
Couche Réseau	@IP src	?
	@IP dest	?
Couches Transport à Application	DATA	

Scénario 1 : M1 envoie un message à M2

M1 et M2 dans le même sous-réseau : envoi d'une trame directement à M2

Trame simplifiée

Couche Liaison	@MAC dest	?
	@MAC src	MAC M1
Couche Réseau	@IP src	192.168.0.1
	@IP dest	?
Couches Transport à Application		DATA

Scénario 1 : M1 envoie un message à M2

M1 et M2 dans le même sous-réseau : envoi d'une trame directement à M2

Trame simplifiée

Couche Liaison	@MAC dest	?
	@MAC src	MAC M1
Couche Réseau	@IP src	192.168.0.1
	@IP dest	192.168.0.2
Couches Transport à Application		DATA

Scénario 1 : M1 envoie un message à M2

M1 et M2 dans le même sous-réseau : envoi d'une trame directement à M2

Trame simplifiée

Couche Liaison	@MAC dest	
	@MAC src	MAC M1
Couche Réseau	@IP src	192.168.0.1
	@IP dest	192.168.0.2
Couches Transport à Application		DATA

Comment connaître l'adresse MAC du destinataire ?

Correspondance IP/MAC

Problème

- Adresse IP au niveau IP et adresse MAC au niveau Ethernet
- Comment les mettre en correspondance ?

ARP : Address Resolution Protocol

- Traduction d'adresse niveau réseau en adresse MAC
- Interface couches OSI 2 (liaison) et 3 (réseau)
- Fonctionnement :
 - Émission d'un broadcast « Requête ARP » sur le réseau local (who-has)
 - Émission d'un unicast « Réponse ARP » contenant l'adresse MAC (is-at)
- Accéléré par un mécanisme de cache (commande arp)
- Représente une vulnérabilité (usurpation)
- Remplacé par ICMPv6 (ICMP enrichi et étendu avec notamment le Neighbor Detector) en IPv6

Scénario 1

M1 et M2 sont dans le même sous-réseau :

- ARP pour obtenir l'adresse MAC de M2

Trame simplifiée

Couche Liaison	@MAC dest	?
	@MAC src	MAC M1
Couche Réseau	@IP src	192.168.0.1
	@IP dest	192.168.0.2
Couches Transport à Application	DATA	

Scénario 1

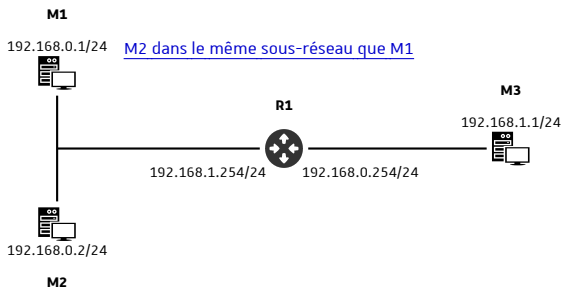
M1 et M2 sont dans le même sous-réseau :

- ARP pour obtenir l'adresse MAC de M2
- envoi d'une trame directement à M2

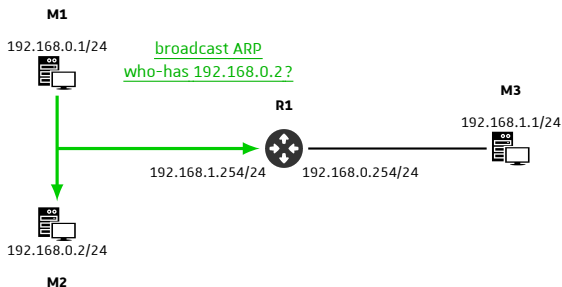
Trame simplifiée

Couche Liaison	@MAC dest	MAC M2
	@MAC src	MAC M1
Couche Réseau	@IP src	192.168.0.1
	@IP dest	192.168.0.2
Couches Transport à Application		DATA

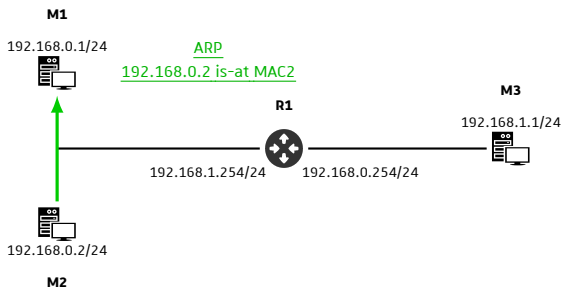
Scénario 1



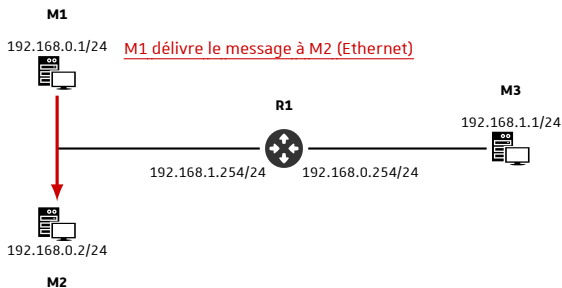
Scénario 1



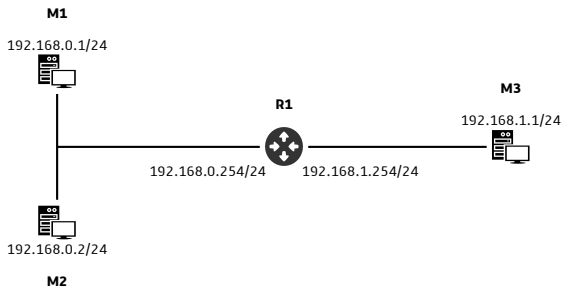
Scénario 1



Scénario 1



Scénario 2 : M1 envoie un message à M3

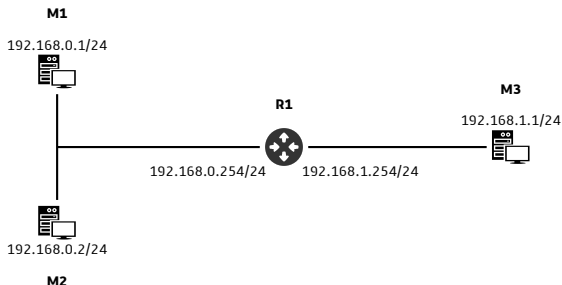


M1 détecte le sous-réseau

- Application du masque de sous-réseau
 - M1 : 192.168.0.1 ET 255.255.255.0 = 192.168.0.0
 - M3 : 192.168.1.1 ET 255.255.255.0 = 192.168.1.0

↪ Les deux machines **ne sont PAS** dans le même sous-réseau

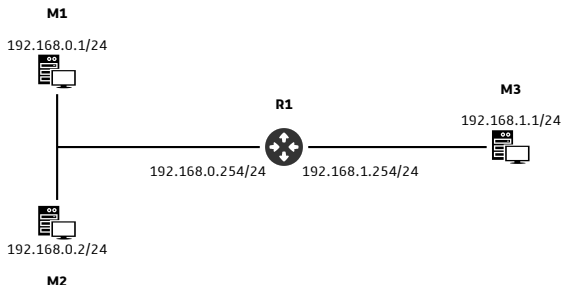
Configuration de l'interface



Configuration de l'interface de M1

• Adresse IP	192.168. 0. 1
• Masque de sous-réseau	255.255.255. 0
• Passerelle	192.168. 0.254

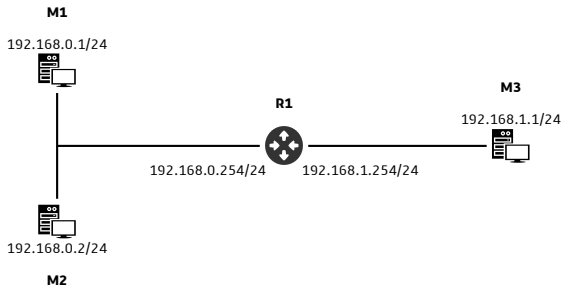
Configuration de l'interface



Configuration de l'interface de M1

- Adresse IP 192.168. 0. 1
- Masque de sous-réseau 255.255.255. 0
- **Passerelle** 192.168. 0.254

Configuration de l'interface



Passerelle

Prochaine destination intermédiaire si la destination finale n'est pas dans le même sous-réseau

Scénario 2

M1 et M3 ne sont PAS dans le même sous-réseau : envoi d'un paquet à M3 via R1

Trame simplifiée

Couche Liaison	@MAC dest	?
	@MAC src	?
Couche Réseau	@IP src	?
	@IP dest	?
Couches Transport à Application	DATA	

Scénario 2

M1 et M3 ne sont PAS dans le même sous-réseau : envoi d'un paquet à M3 via R1

Trame simplifiée

Couche Liaison	@MAC dest	?
	@MAC src	MAC M1
Couche Réseau	@IP src	?
	@IP dest	?
Couches Transport à Application	DATA	

Scénario 2

M1 et M3 ne sont PAS dans le même sous-réseau : envoi d'un paquet à M3 via R1

Trame simplifiée

Couche Liaison	@MAC dest	?
	@MAC src	MAC M1
Couche Réseau	@IP src	192.168.0.1
	@IP dest	?
Couches Transport à Application	DATA	

Scénario 2

M1 et M3 ne sont PAS dans le même sous-réseau : envoi d'un paquet à M3 via R1

Trame simplifiée

Couche Liaison	@MAC dest	?
	@MAC src	MAC M1
Couche Réseau	@IP src	192.168.0.1
	@IP dest	192.168.1.1
Couches Transport à Application	DATA	

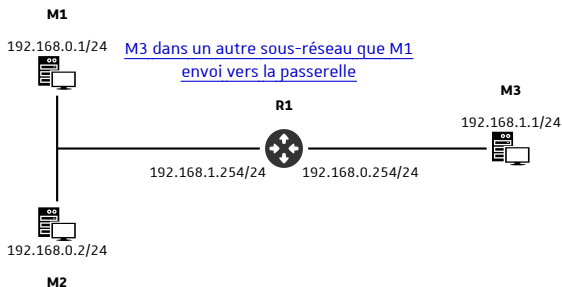
Scénario 2

M1 et M3 ne sont PAS dans le même sous-réseau : envoi d'un paquet à M3 via R1

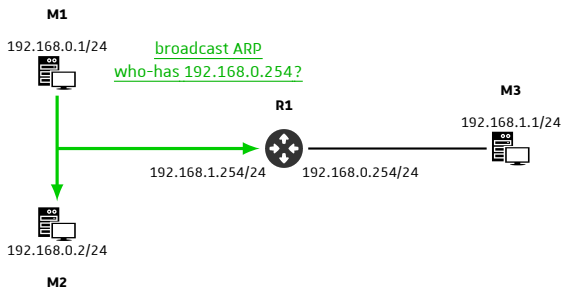
Trame simplifiée

Couche Liaison	@MAC dest	MAC R1
	@MAC src	MAC M1
Couche Réseau	@IP src	192.168.0.1
	@IP dest	192.168.1.1
Couches Transport à Application		DATA

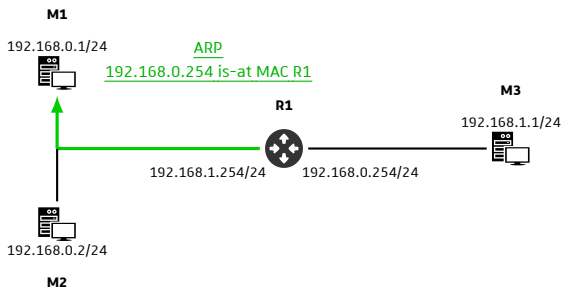
Scénario 2 : M1 envoie un message à M3 via R1



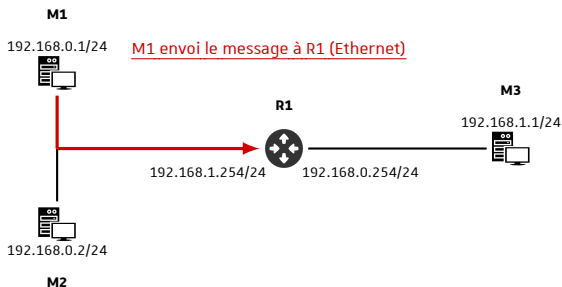
Scénario 2 : M1 envoie un message à M3 via R1



Scénario 2 : M1 envoie un message à M3 via R1



Scénario 2 : M1 envoie un message à M3 via R1



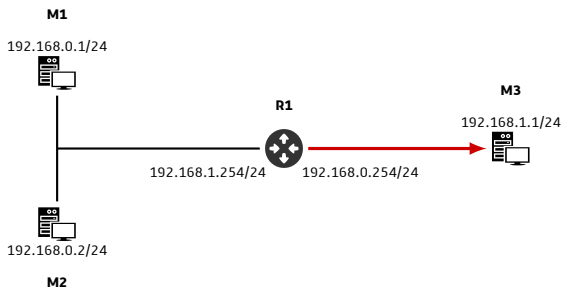
Scénario 2

R1 transmet le paquet IP vers M3

Trame simplifiée

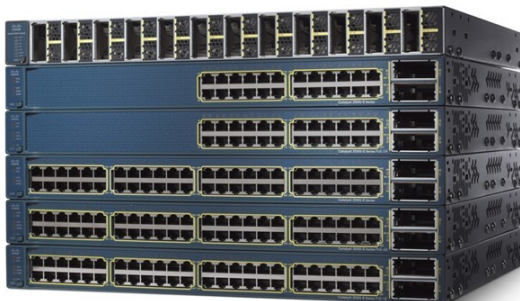
Couche Liaison	@MAC dest	MAC M3
	@MAC src	MAC R1
Couche Réseau	@IP src	192.168.0.1
	@IP dest	192.168.1.1
Couches Transport à Application		DATA

Scénario 2 : R1 transmet la trame à M3



Zoom sur les routeurs

- La fonction des routeurs est de **router les paquets** :
 - c.-à-d. les acheminer au travers du réseau
- Ce sont des machines dédiées au traitement réseau
- Ils possèdent plusieurs interfaces :
 - une interface par sous-réseau connecté



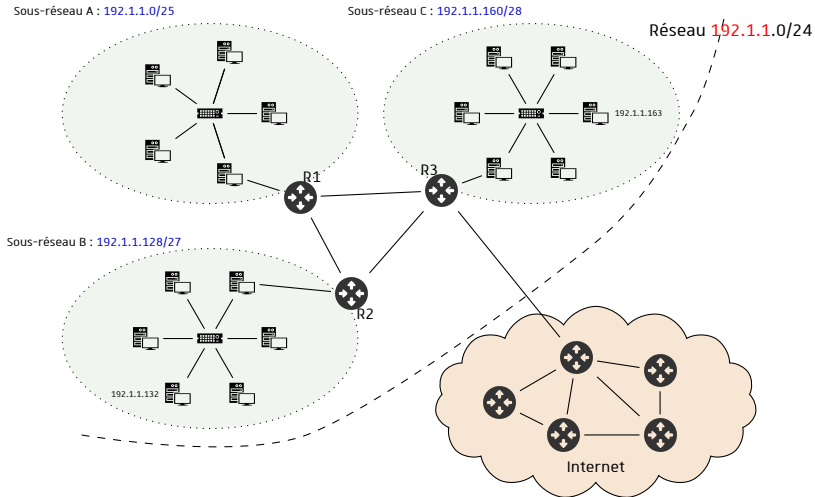
Zoom sur les routeurs

- La fonction des routeurs est de **router les paquets** :
 - c.-à-d. les acheminer au travers du réseau
- Ce sont des machines dédiées au traitement réseau
- Ils possèdent plusieurs interfaces :
 - une interface par sous-réseau connecté
- Ils fonctionnent grâce à une **table de routage**

Table de routage (R1)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1. 0	255.255.255.128	192.1.1. 1	192.1.1. 1
2	192.1.1.128	255.255.255.224	192.1.1.178	192.1.1.177
3	192.1.1.160	255.255.255.240	192.1.1.182	192.1.1.181
4	0.0.0. 0	0. 0. 0. 0	192.1.1.182	192.1.1.181

Routeurs



Routeurs

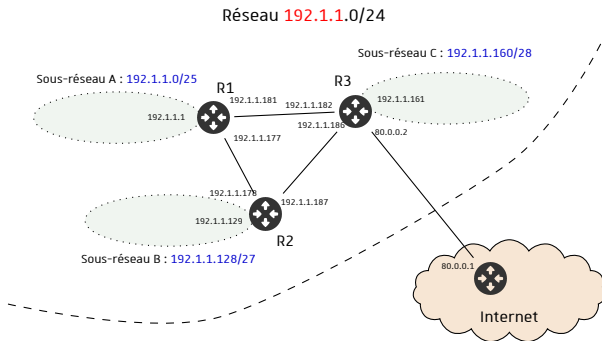


Table de routage (R1)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1. 0	255.255.255.128	192.1.1. 1	192.1.1. 1
2	192.1.1.128	255.255.255.224	192.1.1.178	192.1.1.177
3	192.1.1.160	255.255.255.240	192.1.1.182	192.1.1.181
4	0.0.0. 0	0. 0. 0. 0	192.1.1.182	192.1.1.181

Routeurs

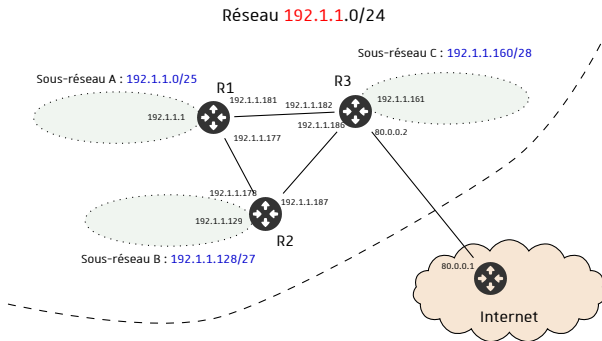


Table de routage (R2)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1. 0	255.255.255.128	192.1.1.177	192.1.1.178
2	192.1.1.128	255.255.255.224	192.1.1.129	192.1.1.129
3	192.1.1.160	255.255.255.240	192.1.1.186	192.1.1.187
4	0.0.0. 0	0. 0. 0. 0	192.1.1.186	192.1.1.187

Routeurs

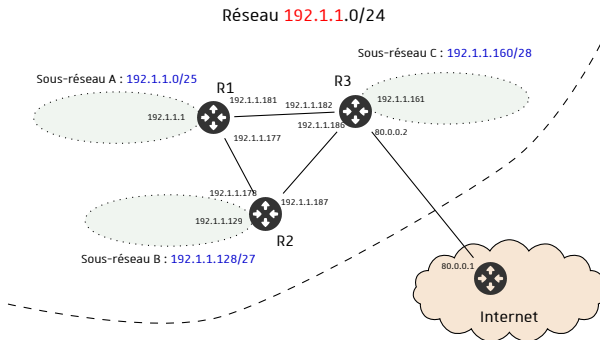


Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1. 0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0. 0	0. 0. 0. 0	80.0.0. 1	80.0.0. 2

Routeurs

Algorithme des routeurs

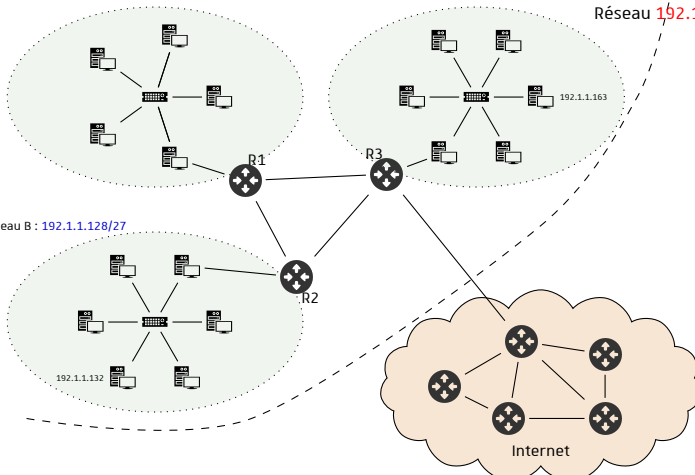
R3 reçoit un paquet à destination de 192.1.1.132

Sous-réseau A : 192.1.1.0/25

Sous-réseau C : 192.1.1.160/28

Réseau 192.1.1.0/24

Sous-réseau B : 192.1.1.128/27



Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.132

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1.0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 192.1.1.132

❶ 192.1.1.132 ET 255.255.255.128 (192.1.1.128) \neq 192.1.1.0

❷ 192.1.1.132 ET 255.255.255.224 (192.1.1.128) = 192.1.1.128

⇒ Le paquet est envoyé vers 192.1.1.187 *via* 192.1.1.186

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.132

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1.0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 192.1.1.132

- 1 192.1.1.132 ET 255.255.255.128 (192.1.1.128) \neq 192.1.1.0
- 2 192.1.1.132 ET 255.255.255.224 (192.1.1.128) = 192.1.1.128

⇒ Le paquet est envoyé vers **192.1.1.187** *via* **192.1.1.186**

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.132

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1.0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 192.1.1.132

- 1 192.1.1.132 ET 255.255.255.128 (192.1.1.128) \neq 192.1.1.0
- 2 192.1.1.132 ET 255.255.255.224 (192.1.1.128) = 192.1.1.128

⇒ Le paquet est envoyé vers **192.1.1.187** *via* **192.1.1.186**

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.132

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1.0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 192.1.1.132

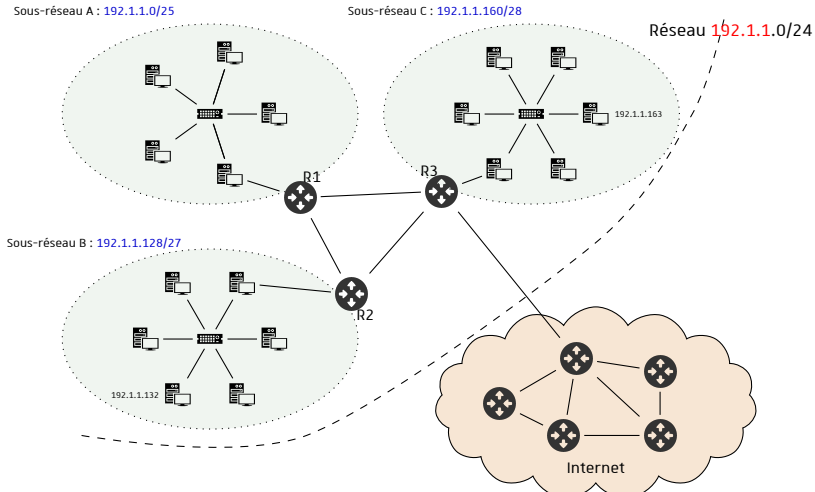
- 1 192.1.1.132 ET 255.255.255.128 (192.1.1.128) \neq 192.1.1.0
- 2 192.1.1.132 ET 255.255.255.224 (192.1.1.128) = 192.1.1.128

⇒ Le paquet est envoyé vers **192.1.1.187** *via* **192.1.1.186**

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.163



Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.163

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1.0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 192.1.1.163

- ❶ 192.1.1.163 ET 255.255.255.128 (192.1.1.128) \neq 192.1.1.0
- ❷ 192.1.1.163 ET 255.255.255.224 (192.1.1.160) \neq 192.1.1.128
- ❸ 192.1.1.163 ET 255.255.255.160 (192.1.1.160) = 192.1.1.160

⇒ @Passerelle = @Interface donc on peut envoyer directement à la destination

⇒ La trame est envoyée vers 192.1.1.163 *via* 192.1.1.161 (cf. ARP)

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.163

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1.0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 192.1.1.163

- ❶ 192.1.1.163 ET 255.255.255.128 (192.1.1.128) \neq 192.1.1.0
- ❷ 192.1.1.163 ET 255.255.255.224 (192.1.1.160) \neq 192.1.1.128
- ❸ 192.1.1.163 ET 255.255.255.160 (192.1.1.160) = 192.1.1.160

⇒ @Passerelle = @Interface donc on peut envoyer directement à la destination

⇒ La trame est envoyée vers 192.1.1.163 *via* 192.1.1.161 (cf. ARP)

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.163

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1.0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 192.1.1.163

- 1 192.1.1.163 ET 255.255.255.128 (192.1.1.128) \neq 192.1.1.0
- 2 192.1.1.163 ET 255.255.255.224 (192.1.1.160) \neq 192.1.1.128
- 3 192.1.1.163 ET 255.255.255.160 (192.1.1.160) = 192.1.1.160

⇒ @Passerelle = @Interface donc on peut envoyer directement à la destination

⇒ La trame est envoyée vers **192.1.1.163** *via* **192.1.1.161** (cf. ARP)

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.163

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1.0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 192.1.1.163

- ❶ 192.1.1.163 ET 255.255.255.128 (192.1.1.128) \neq 192.1.1.0
- ❷ 192.1.1.163 ET 255.255.255.224 (192.1.1.160) \neq 192.1.1.128
- ❸ 192.1.1.163 ET 255.255.255.160 (192.1.1.160) = 192.1.1.160

⇒ @Passerelle = @Interface donc on peut envoyer directement à la destination

⇒ La trame est envoyée vers **192.1.1.163** *via* **192.1.1.161** (cf. ARP)

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.163

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1.0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 192.1.1.163

- 1 192.1.1.163 ET 255.255.255.128 (192.1.1.128) \neq 192.1.1.0
- 2 192.1.1.163 ET 255.255.255.224 (192.1.1.160) \neq 192.1.1.128
- 3 192.1.1.163 ET 255.255.255.160 (192.1.1.160) = 192.1.1.160

⇒ @Passerelle = @Interface donc on peut envoyer directement à la destination

⇒ La trame est envoyée vers **192.1.1.163** *via* **192.1.1.161** (cf. ARP)

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 192.1.1.163

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1.0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 192.1.1.163

- 1 192.1.1.163 ET 255.255.255.128 (192.1.1.128) \neq 192.1.1.0
- 2 192.1.1.163 ET 255.255.255.224 (192.1.1.160) \neq 192.1.1.128
- 3 192.1.1.163 ET 255.255.255.160 (192.1.1.160) = 192.1.1.160

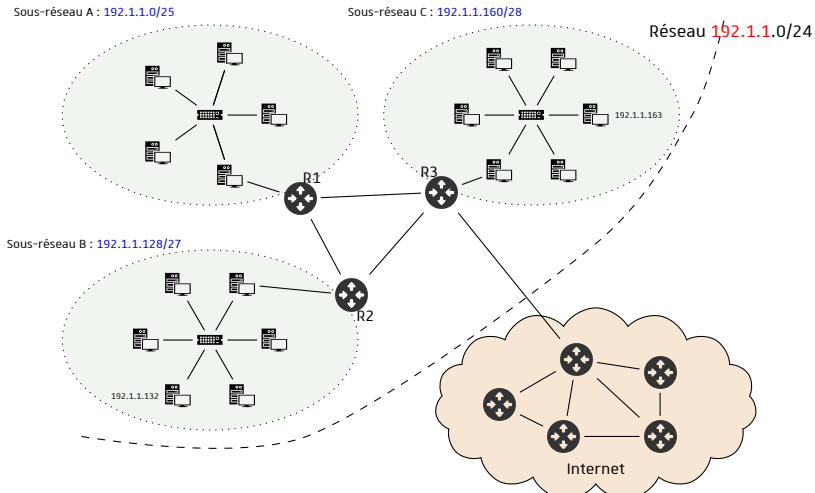
⇒ @Passerelle = @Interface donc on peut envoyer directement à la destination

⇒ La trame est envoyée vers **192.1.1.163** *via* **192.1.1.161** (cf. ARP)

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 132.2.5.1



Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 132.2.5.1

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1. 0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 132.2.5.1

- ❶ 132.2.5.1 ET 255.255.255.128 (132.2.5.0) \neq 192.1.1.0
- ❷ 132.2.5.1 ET 255.255.255.224 (132.2.5.0) \neq 192.1.1.128
- ❸ 132.2.5.1 ET 255.255.255.160 (132.2.5.0) \neq 192.1.1.160
- ❹ 132.2.5.1 ET 0. 0. 0. 0 (0.0.0.0) = 0.0.0. 0

⇒ Le paquet est envoyé vers **80.0.0.1** *via* **192.1.1.254**

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 132.2.5.1

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1. 0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 132.2.5.1

- 1 132.2.5.1 ET 255.255.255.128 (132.2.5.0) \neq 192.1.1.0
- 2 132.2.5.1 ET 255.255.255.224 (132.2.5.0) \neq 192.1.1.128
- 3 132.2.5.1 ET 255.255.255.160 (132.2.5.0) \neq 192.1.1.160
- 4 132.2.5.1 ET 0. 0. 0. 0 (0.0.0.0) = 0.0.0. 0

⇒ Le paquet est envoyé vers **80.0.0.1** *via* **192.1.1.254**

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 132.2.5.1

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1. 0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 132.2.5.1

- 1 132.2.5.1 ET 255.255.255.128 (132.2.5.0) \neq 192.1.1.0
- 2 132.2.5.1 ET 255.255.255.224 (132.2.5.0) \neq 192.1.1.128
- 3 132.2.5.1 ET 255.255.255.160 (132.2.5.0) \neq 192.1.1.160
- 4 132.2.5.1 ET 0. 0. 0. 0 (0.0.0.0) = 0.0.0. 0

⇒ Le paquet est envoyé vers **80.0.0.1** *via* **192.1.1.254**

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 132.2.5.1

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1. 0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 132.2.5.1

- 1 132.2.5.1 ET 255.255.255.128 (132.2.5.0) \neq 192.1.1.0
- 2 132.2.5.1 ET 255.255.255.224 (132.2.5.0) \neq 192.1.1.128
- 3 132.2.5.1 ET 255.255.255.160 (132.2.5.0) \neq 192.1.1.160
- 4 132.2.5.1 ET 0. 0. 0. 0 (0.0.0.0) = 0.0.0. 0

⇒ Le paquet est envoyé vers **80.0.0.1** *via* **192.1.1.254**

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 132.2.5.1

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1. 0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 132.2.5.1

- 1 132.2.5.1 ET 255.255.255.128 (132.2.5.0) \neq 192.1.1.0
- 2 132.2.5.1 ET 255.255.255.224 (132.2.5.0) \neq 192.1.1.128
- 3 132.2.5.1 ET 255.255.255.160 (132.2.5.0) \neq 192.1.1.160
- 4 132.2.5.1 ET 0. 0. 0. 0 (0.0.0.0) = 0.0.0. 0

⇒ Le paquet est envoyé vers **80.0.0.1** *via* **192.1.1.254**

Routeurs

Algorithme des routeurs

R3 reçoit un paquet à destination de 132.2.5.1

Table de routage (R3)

N	Adr Dest	Masque	Passerelle	Interface
1	192.1.1. 0	255.255.255.128	192.1.1.181	192.1.1.182
2	192.1.1.128	255.255.255.224	192.1.1.187	192.1.1.186
3	192.1.1.160	255.255.255.240	192.1.1.161	192.1.1.161
4	0.0.0.0	0.0.0.0	80.0.0.1	192.1.1.254

Si (Adresse IP ET Masque = Adr Dest)

Alors Envoyer le paquet vers Passerelle *via* Interface

À la réception par R3 d'un paquet à destination de 132.2.5.1

- 1 132.2.5.1 ET 255.255.255.128 (132.2.5.0) \neq 192.1.1.0
- 2 132.2.5.1 ET 255.255.255.224 (132.2.5.0) \neq 192.1.1.128
- 3 132.2.5.1 ET 255.255.255.160 (132.2.5.0) \neq 192.1.1.160
- 4 132.2.5.1 ET 0. 0. 0. 0 (0.0.0.0) = 0.0.0. 0

⇒ Le paquet est envoyé vers **80.0.0.1** *via* **192.1.1.254**

Dans l'Internet d'aujourd'hui, en pratique :

- chaque routeur (re)construit en permanence sa table de routage
- un paquet est routé au saut par saut (hop by hop) : interrogation des tables de routage successives
 - aucun routeur ne connaît la route exacte empruntée par un paquet.
Risque de boucle?

Si changement dans la table de routage :

- 2 paquets successifs (même flux $\langle @IP_{SRC}, @IP_{DST}, port_{SRC}, port_{DST} \rangle$)

⇒ Éventuellement routes différentes

Problème

Comment construire les tables de routage?

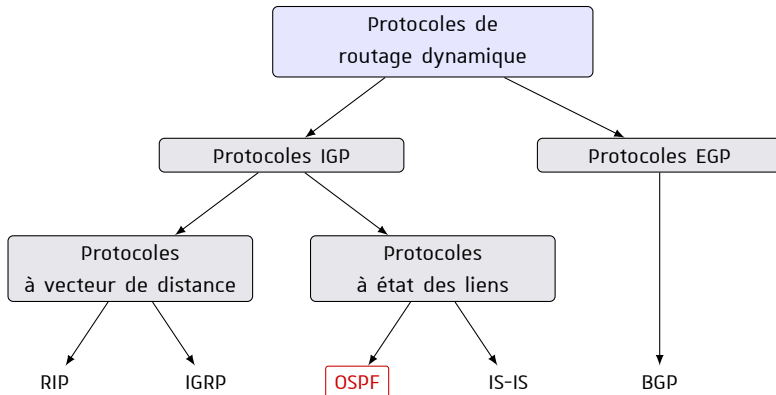
Algorithme de routage

Objectifs

Permettre la construction des tables de routage tel que les actions suivantes soient « garanties » :

- assurer l'acheminement des paquets (livraison à destination → pas de boucle)
- minimiser les délais de transmission
- maximiser les débits
- prévenir les congestions
- assurer l'équité entre utilisateurs (ou QoS)
- optimiser l'utilisation des ressources du réseau
- gérer les défaillances du réseau (pannes liens/routeurs...)
- s'adapter aux modifications du réseau (topologique + charge?)
- stabilité (éviter les oscillations)

La grande famille des protocoles de routage



Protocoles de routage

Système autonome, routage intérieur et extérieur

- Au sein d'un système autonome (AS), les routes sont générées par des protocoles de routage **intérieurs** (RIP, IGRP, EIGRP, OSPF ou IS-IS)
- Les protocoles de routage qui permettent de connecter les AS entre eux sont des protocoles de routage **extérieurs** (EGP ou BGP)
- Protocoles à vecteur de distance :
 - basés sur l'échange des tables de routage entre routeurs
 - addition des distances pour trouver les routes (Bellman-Ford)
 - sensibles aux boucles de routage
 - convergence lente
- Protocoles à état de liens :
 - algorithme plus efficace (Dijkstra ou Shortest Path First)
 - les routeurs collectent l'ensemble des coûts des liens et construisent de leur point de vue l'arbre de tous les chemins
 - les meilleures routes sont intégrées à la table de routage
 - convergence rapide

Routage par informations d'état des liens (*link state routing*)

→ **chaque routeur possède une vue globale du réseau**

- OSPF : Open Shortest Path First
 - v2 RFC 2328 en 1998 (IPv4)
 - v3 RFC 5340 en 2008 (IPv6)
- IS-IS : Intermediate System to Intermediate System
 - v1 RFC 1142 en 1990 (indépendant du protocole réseau)

Routage à état des liens

- Tous les routeurs possèdent à un instant donné la même table des liens (LSBD)
- Représentation globale du réseau sous la forme d'un graphe connu de tous :
 - arête du graphe : lien = connexion entre routeur et réseau ou entre 2 routeurs
 - sommet du graphe : routeurs et réseaux

Principe

- 1 Chaque routeur détermine l'état des liens (LS) adjacents
- 2 Si changement état d'un lien local → diffusion de proche en proche du nouvel LS à l'ensemble du réseau (*flooding*)

Note : tous les routeurs ont la même LSDB au délai de diffusion des LS près

À la réception d'un LSA

À chaque réception de LS Advertisement (LSA), un routeur R calcule le plus court chemin de lui vers le reste du monde :

- Arbre des plus court chemins (SPT) calculé avec Dijkstra (poids positifs)
- ex-aequo départagé par lecture lexicographique des id. (ou ECMP)
- racine du SPT = routeur R

⇒ Prochain saut vers $@IP_{DST}$ = premier fils conduisant à $@IP_{DST}$ dans le SPT (sous-arbre contenant $@IP_{DST}$)

OSPF – RFC 2328

Open Shortest Path First

Caractéristiques

- Protocole à état des liens
- Intra-domaine : routage interne à un réseau (ou AS, ou domaine) :
 - entreprise, réseaux locaux étendu (p.ex. OSIRIS), FAI...
- Découverte automatique des voisins
- Élection d'un routeur désigné par sous-réseau (LAN)
- Diffusion fiable des LSA
- Possibilité de découper le réseau en aires (OSPF area) :
 - hiérarchisation pour les grands réseaux

OSPF - RFC 2328

Déroulement complet d'OSPF

Chaque routeur :

- découvre son voisinage et conserve une liste de tous ses voisins (NT);
- utilise un protocole fiable pour échanger les informations topologiques avec ses voisins (LSA)
- stocke les informations topologiques apprises dans sa base de données (LSBD)
- exécute l'algorithme SPF (Dijkstra) pour calculer les meilleurs routes
- place ensuite la meilleure route vers chaque sous-réseau dans sa table de routage (RT)

Chaque routeur possède :

- une table de ses voisins, Neighbor table (NT)
- une base de données de la topologie du réseau, Topology database (LSBD)
- une table de routage, Routing table (RT)

L'essentiel

- Principe d'encapsulation : paquet IP est encapsulé dans une trame Ethernet
- ARP : correspondance @MAC/@IP
- Routeurs : machines dédiées aux traitements réseau
- Routeurs utilisent une table de routage pour savoir comment acheminer les paquets
- Table de routage construites manuellement ou en utilisant un algorithme de routage
- Routage à état des liens permet de construire automatiquement les tables de routage