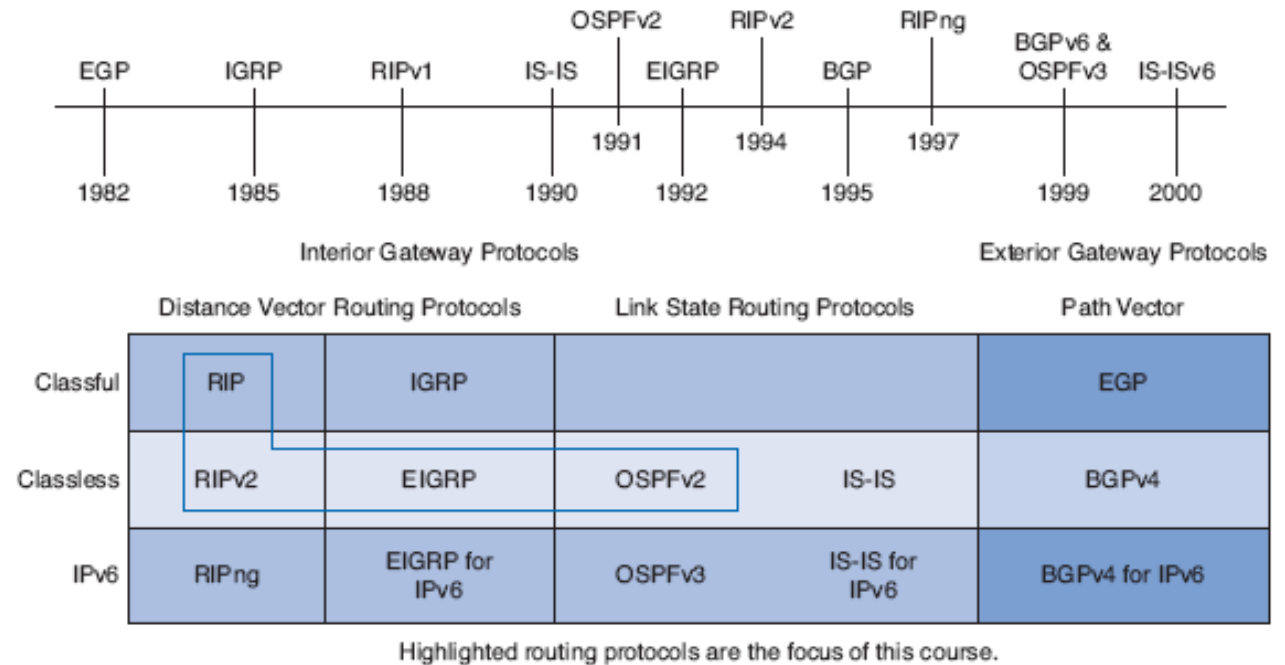


# Interconnexion à grande échelle

- VLSM et CIDR
- Les protocoles par Vecteur de Distance (suite)
  - Le protocole RIP version 2
  - EIGRP

# Protocoles Classful et Classless



Deux types de protocole de routage :

classful ou classless

C'est le résultat de l'évolution du système d'adressage IP

Dès que les réseaux ont adopté des adressages classless, les protocoles de routage ont du s'adapter afin d'inclure les masques dans les mises à jour

# Comment les adresses étaient attribuées

---

RFC 790 :

"The assignment of numbers is also handled by Jon. If you are developing a protocol or application that will require the use of a link, socket, port, protocol, or network number please contact Jon to receive a number assignment.

Jon Postel

USC - Information Sciences Institute

4676 Admiralty Way

Marina del Rey, California 90291"

Plus sérieusement, les blocs d'adresses étaient distribués sans trop de contrôle au début...

Jusqu'à la crise des adresses des années 1990

# Adressage par Classes IP

Class A	Network		Host	
Octet	1	2	3	4

Class B	Network		Host	
Octet	1	2	3	4

Class C	Network			Host
Octet	1	2	3	4

Class D	Host			
Octet	1	2	3	4

Address Class	First Octet Range	Number of Possible Networks	Number of Hosts per Network
Class A	0 to 127	128 (2 are reserved)	16,777,214
Class B	128 to 191	16,348	65,534
Class C	192 to 223	2,097,152	254

- Au début d'Internet, les adresses IP étaient attribuées aux entreprises selon leur demande au lieu de leur besoin réel (ou futur)
- Lorsqu'un réseau était attribué, cette adresse était associée à une "**Classe**", **A, B, ou C**.
- Le premier octet de l'adresse détermine quelle est sa classe et comment les parties réseau et hôte sont divisées
- Ce n'est qu'en 1992 que l'IETF introduit le CIDR (Classless Inter-Domain Routing), ce qui rend les classes obsolètes
  - Aussi connu comme **adressage IP Classless**
- Actuellement l'Internet est classless, mais certains protocoles anciens conservent encore un comportement par classes



# Premiers bits ?

	1st Octet	2nd Octet	3rd Octet	4th Octet	Subnet Mask
<b>Class A</b>	Network	Host	Host	Host	255.0.0.0 or /8
<b>Class B</b>	Network	Network	Host	Host	255.255.0.0 or /16
<b>Class C</b>	Network	Network	Network	Host	255.255.255.0 or /24

Class	High-Order Bits	Start	End
Class A		0.0.0.0	127.255.255.255
Class B		128.0.0.0	191.255.255.255
Class C		192.0.0.0	223.255.255.255
Multicast		224.0.0.0	239.255.255.255
Experimental		240.0.0.0	255.255.255.255

Le modèle par classes utilisait seulement trois tailles de réseau

Comment ceci a été établi ?

Comment un dispositif réseau ancien pouvait trouver rapidement la classe d'une adresse et déterminer son masque ?

Il suffisait d'examiner les premiers bits de l'adress



# Subnetting

$2^{10}$	$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
1,024	512	256	128	64	32	16	8	4	2	1
Number of bits borrowed:			Number of hosts per subnet							
10	9	8	7	6	5	4	3	2	1	
1,024	512	256	128	64	32	16	8	4	2	1
						Hosts or Subnets		Number of subnets		

172.16.1. 0 0 0 0 0 0 0 0  
 255.255.255. 1 1 1 0 0 0 0 0 255.255.255.224  
 3 subnet bits →  
 Network Host

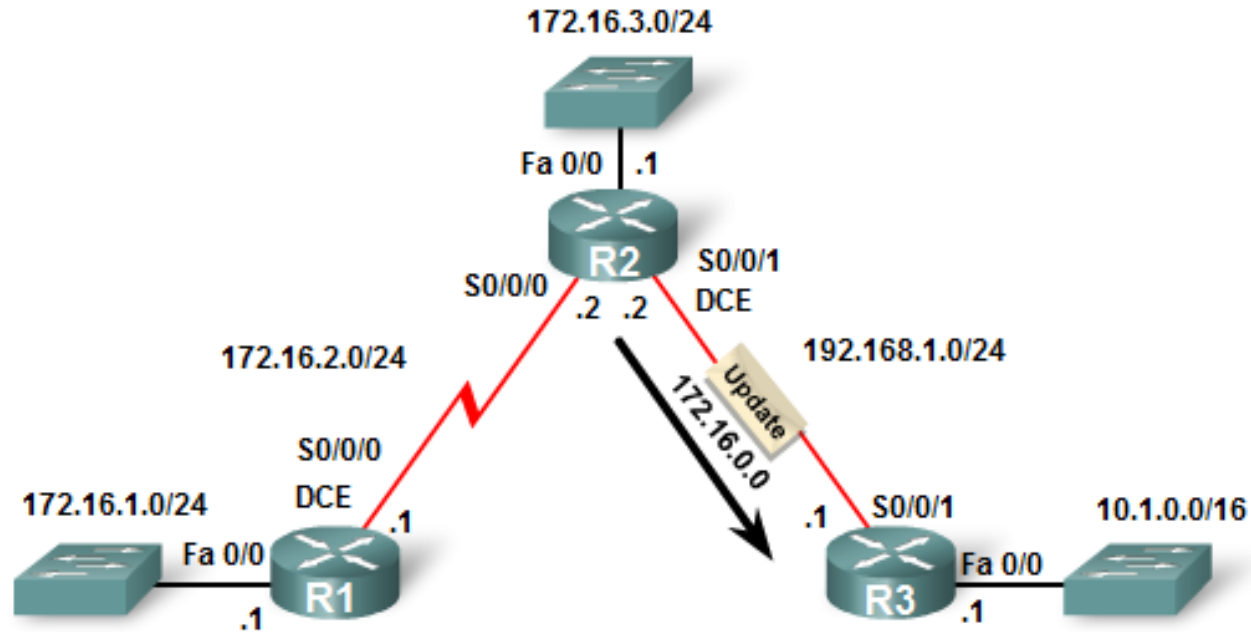
## Masques de sous-réseaux

Pour mieux gérer une classe attribuée, on peut sous-diviser le réseau

Formalisés en 1985 (RFC 950), les masques permettent la division des classes A, B ou C en sous-réseaux plus petits

Ceci permet une utilisation plus rationnelle (moins de gaspillage) et une redistribution des adresses disponibles

# Protocoles de Routage Classful



Protocoles de routage Classful

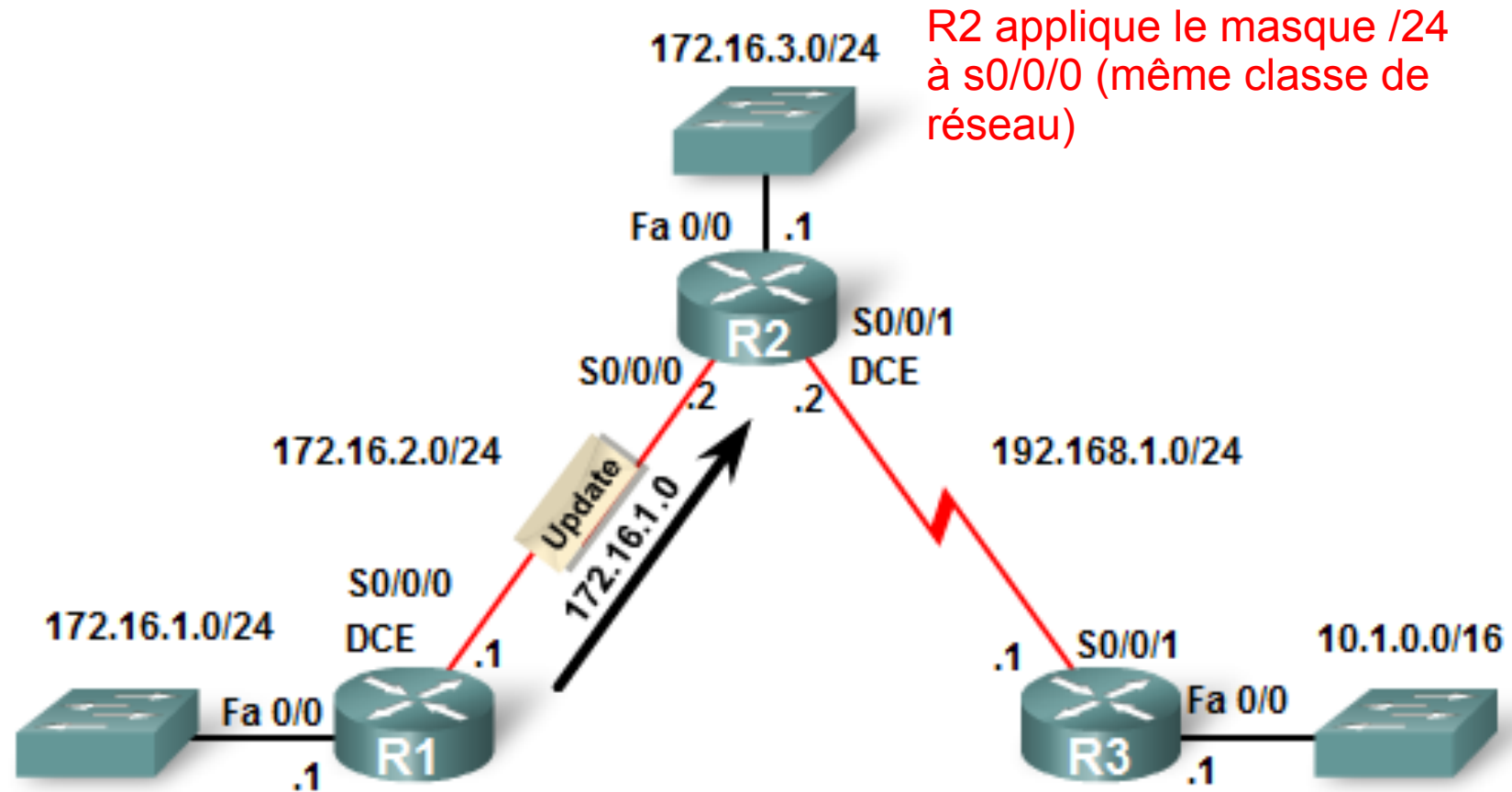
Est-ce que les masques sont inclus dans les messages de mise à jour ?

Non

Alors comment un routeur déterminait le masque ?

à partir de la valeur du premier octet (ou plutôt les trois premiers bits)

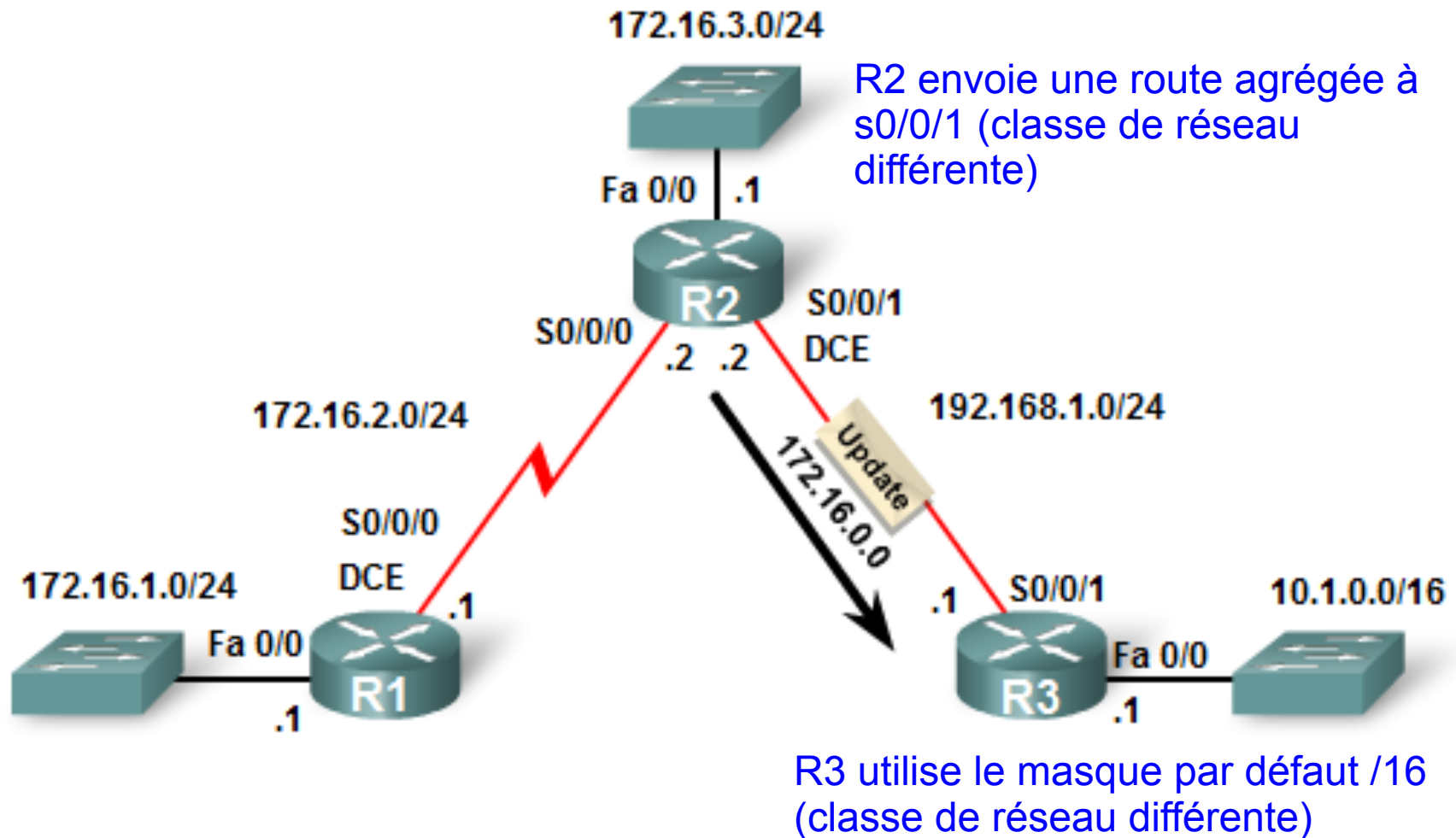
# Protocoles de Routage Classful



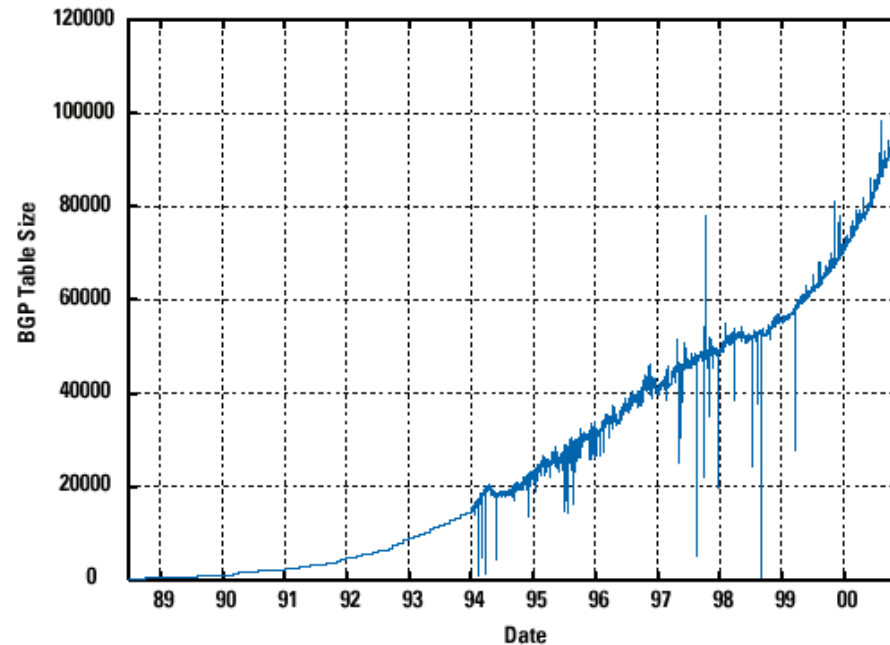
R2 applique le masque /24 à s0/0/0 (même classe de réseau)

R1 envoie une adresse à s0/0/0 (même classe de réseau)

# Protocoles de Routage Classful



# Adressage Classful



L'adressage classful a été la règle jusqu'au début des années 1990, lors de la première crise d'Internet. Si IPv4 est encore utilisé, c'est notamment à cause de l'association de plusieurs techniques :

**VLSM** - 1993 (RFC 1519)

**CIDR** - 1993 (RFC 1519) – agrégation de préfixes

Network Address Translation (**NAT**) - 1994 (RFC 1631)

**Private addressing** - 1996 (RFC 1918)

# Solutions à plus long terme

- Si la première crise de IPv4 a été soulagée grâce à l'utilisation des masques et du NAT, des solutions plus robustes ont été proposées
- Le protocole IPv6 utilise des adresses à 128-bits, ce qui signifie
- 340 282 366 920 938 463 463 374 607 431 768 211 456 adresses
- La mise en place de IPv6 est plutôt lente
  - IPv6 requiert la reconfiguration du matériel réseau et l'entraînement du personnel technique
- L'épuisement « complet » de IPv4 semble forcer une mise en place plus rapide
  - Les systèmes d'exploitation actuels travaillent nativement avec IPv6
  - La configuration du matériel réseau est plutôt simple à faire
  - Les FAI commencent à attribuer des adresses IPv6 (Free.fr, etc) avec des avantages considérables

**11111111.00000000.00000000.00000000 /8 (255.0.0.0) 16,777,216 host addresses CLASSE A**

11111111.10000000.00000000.00000000 /9 (255.128.0.0) 8,388,608 host addresses

11111111.11000000.00000000.00000000 /10 (255.192.0.0) 4,194,304 host addresses

11111111.11100000.00000000.00000000 /11 (255.224.0.0) 2,097,152 host addresses

11111111.11110000.00000000.00000000 /12 (255.240.0.0) 1,048,576 host addresses

11111111.11111000.00000000.00000000 /13 (255.248.0.0) 524,288 host addresses

11111111.11111100.00000000.00000000 /14 (255.252.0.0) 262,144 host addresses

11111111.11111110.00000000.00000000 /15 (255.254.0.0) 131,072 host addresses

**11111111.11111111.00000000.00000000 /16 (255.255.0.0) 65,536 host addresses CLASSE B**

11111111.11111111.10000000.00000000 /17 (255.255.128.0) 32,768 host addresses

11111111.11111111.11000000.00000000 /18 (255.255.192.0) 16,384 host addresses

11111111.11111111.11100000.00000000 /19 (255.255.224.0) 8,192 host addresses

11111111.11111111.11110000.00000000 /20 (255.255.240.0) 4,096 host addresses

11111111.11111111.11111000.00000000 /21 (255.255.248.0) 2,048 host addresses

11111111.11111111.11111100.00000000 /22 (255.255.252.0) 1,024 host addresses

11111111.11111111.11111110.00000000 /23 (255.255.254.0) 512 host addresses

**11111111.11111111.11111111.00000000 /24 (255.255.255.0) 256 host addresses CLASSE C**

11111111.11111111.11111111.10000000 /25 (255.255.255.128) 128 host addresses

11111111.11111111.11111111.11000000 /26 (255.255.255.192) 64 host addresses

11111111.11111111.11111111.11100000 /27 (255.255.255.224) 32 host addresses

11111111.11111111.11111111.11110000 /28 (255.255.255.240) 16 host addresses

11111111.11111111.11111111.11111000 /29 (255.255.255.248) 8 host addresses

11111111.11111111.11111111.11111100 /30 (255.255.255.252) 4 host addresses

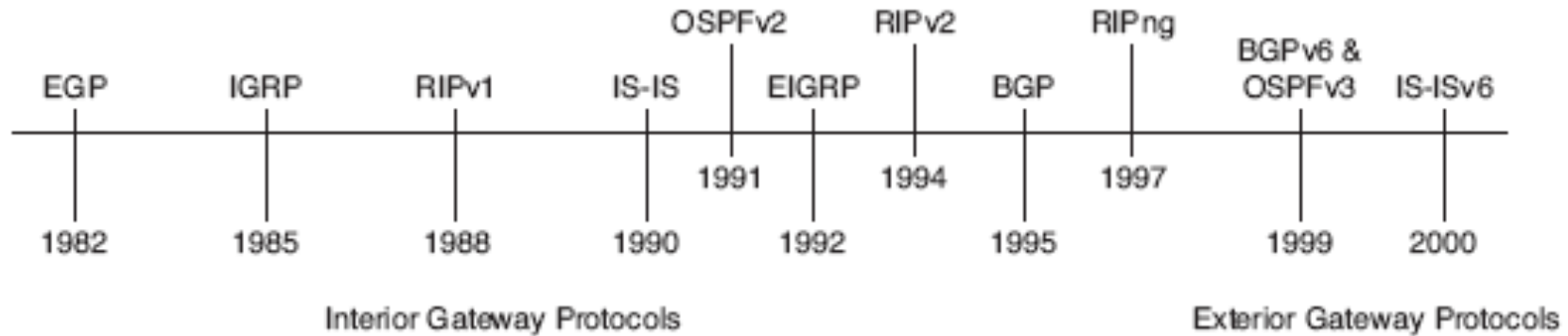
11111111.11111111.11111111.11111110 /31 (255.255.255.254) 2 host addresses

11111111.11111111.11111111.11111111 /32 (255.255.255.255) "Host Route"

Les FAI peuvent désormais  
utilise plus que trois masques  
– adéquation au nombre de  
machines



# Protocoles Classless



	Distance Vector Routing Protocols	Link State Routing Protocols		Path Vector
Classful	RIP	IGRP		EGP
Classless	RIPv2	EIGRP	OSPFv2	BGPv4
IPv6	RIPv2	EIGRP for IPv6	OSPFv3	BGPv4 for IPv6

Highlighted routing protocols are the focus of this course.

Les protocoles classless incluent le masque dans leurs messages

Aujourd'hui, pratiquement tous les protocoles classful ont disparu, sauf RIPv1

# Masque de sous-réseau

Use the following IP address:

IP address:

192 . 168 . 1 . 100

Subnet mask:

255 . 255 . 255 . 0

**Machine: “Je suis une machine dans le réseau 192.168.1.0/24.”**

- Ok, on sait que le masque est utilisé pour séparer la partie réseau de la partie hôte d'une adresse
- Dans une machine, le masque indique quel est le réseau auquel la machine appartient.
- **Pourquoi la machine a besoin de savoir quel est son réseau ?**

# Masque de sous-réseau

Use the following IP address:

IP address:

192 . 168 . 1 . 100

Subnet mask:

255 . 255 . 255 . 0

**Machine: “Je suis une machine dans le réseau 192.168.1.0/24.”**

- **Pourquoi la machine a besoin de savoir quel est son réseau ?**
- Car c'est ainsi qu'elle décide comment encapsuler le paquet IP dans une trame Ethernet :
  - Soit l'adresse MAC de la passerelle par défaut
    - Pour cela il faut connaître l'adresse IP de la passerelle
  - Soit l'adresse MAC de la machine destination, si cela se trouve dans le même sous-réseau

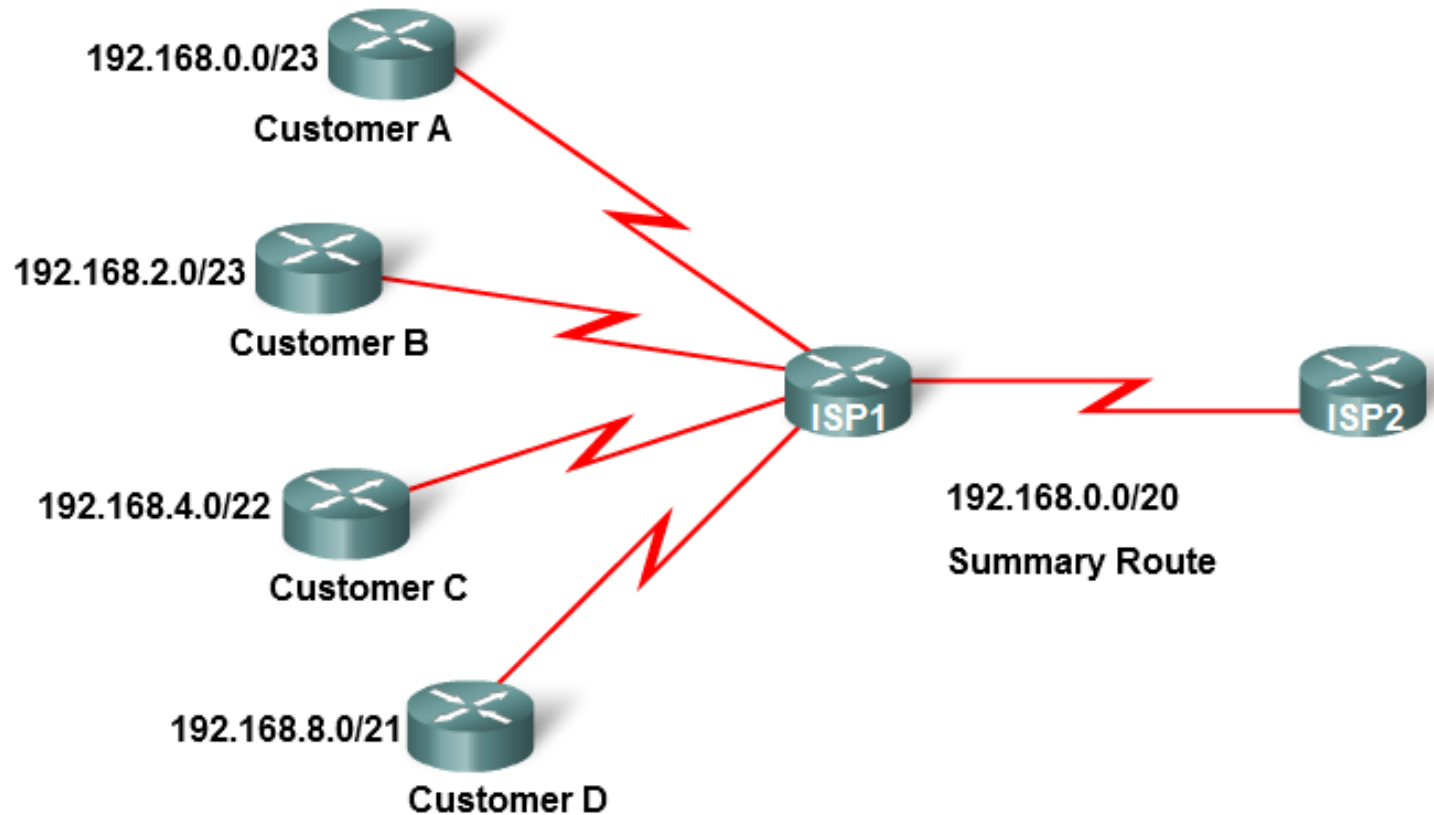
# Masque de sous-réseau

	Réseau	Hôte
IP :	172.16.33.10	10101100.00010000.00100001.000001010
Masque :	255.255.0.0	11111111.11111111.00000000.00000000
	-----	-----
Net Add:	172.16.0.0	10101100.00010000.00000000.00000000

- Pour déterminer si une adresse est dans notre réseau, on fait un ET entre
  - L'adresse destination
  - Notre masque de sous-réseau
- Rappel : l'opération ET
  - 1 ET 1 = 1
  - 0 ET \* = 0



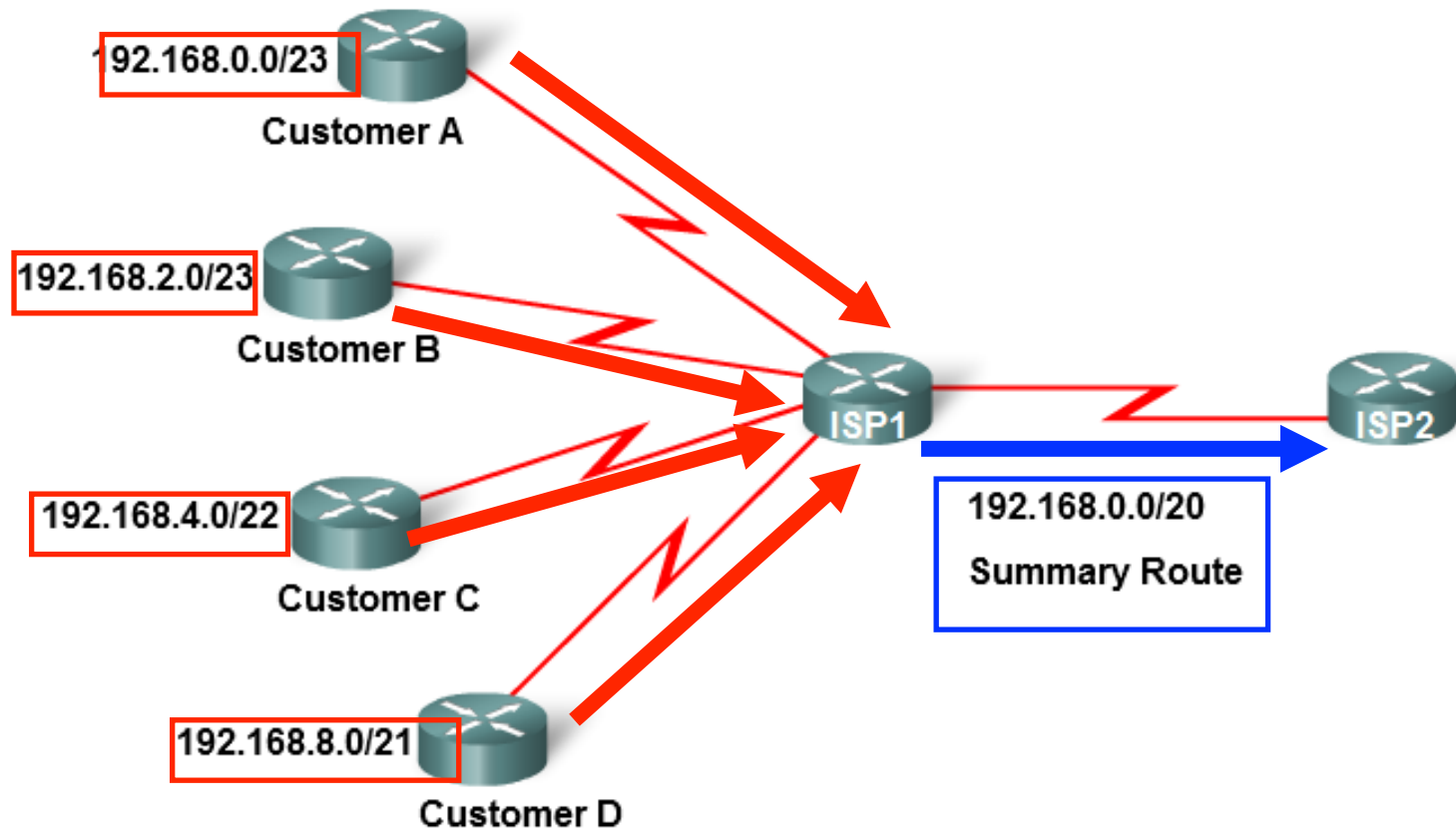
# CIDR et l'Agrégation de routes



CIDR = Agrégation de routes

Un **supernet** regroupe plusieurs réseaux avec un masque plus petit que celui de la classe classful

# CIDR et l'Agrégation de routes

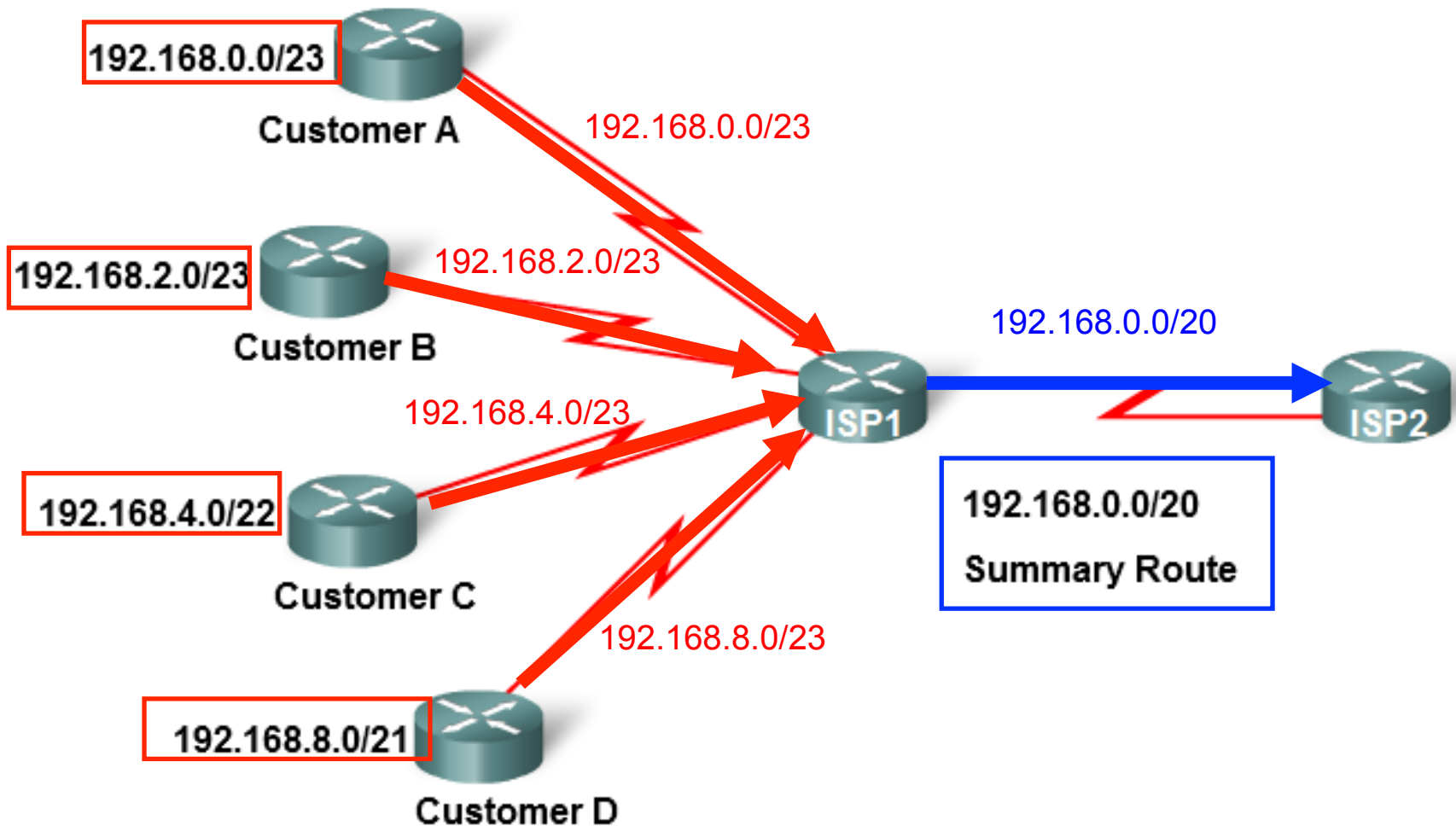


192.168.0.0/23, 192.168.2.0/23, 192.168.4.0/22, and 192.168.8.0/21 sont tous des sous-réseaux individuels envoyés à ISP1

192.168.0.0/23, 192.168.2.0/23, 192.168.4.0/22, and 192.168.8.0/21 sont des sous-réseaux de 192.168.0.0/20

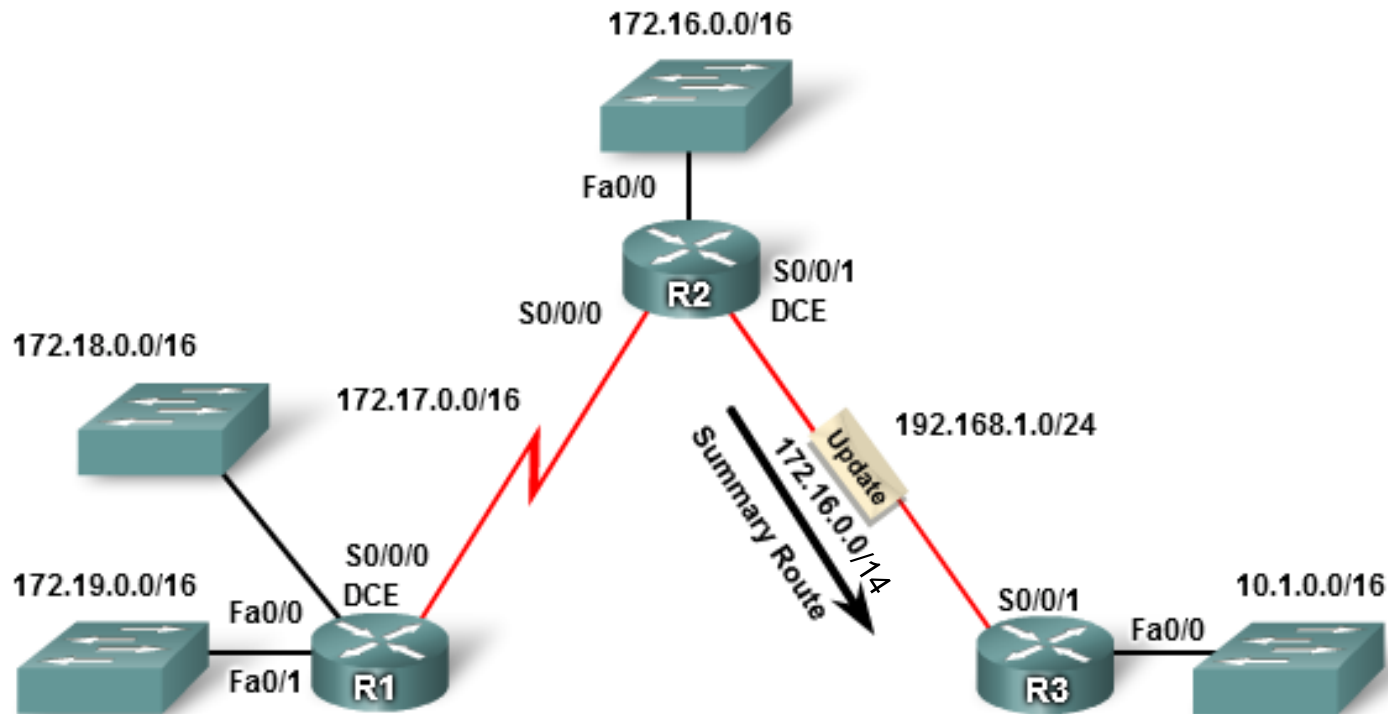
192.168.0.0/20 est une seule route agrégée qui regroupe ces trois réseaux et envoyée à ISP2.

# CIDR et l'Agrégation de routes



La propagation du VLSM et des nécessite un protocole de routage classless car le masque ne peut plus être déterminé par la valeur du premier octet

# Protocoles de Routage Classless



172.16.0.0/16, 172.17.0.0/16, 172.18.0.0/16, et 172.19.0.0/16 peuvent être agrégés dans le réseau 172.16.0.0/14.

Ce réseau **Supernet** /14 (255.252.0.0) est inclus dans les mises à jour



# VLSM – Variable Length Subnet Mask

- VLSM en action
- VLSM et les adresses IP

# VLSM

Le réseau 10.0.0.0/8 a été divisé en sous-réseaux /16, ce qui donne 256 sous-réseaux :

10.0.0.0/16

10.1.0.0/16

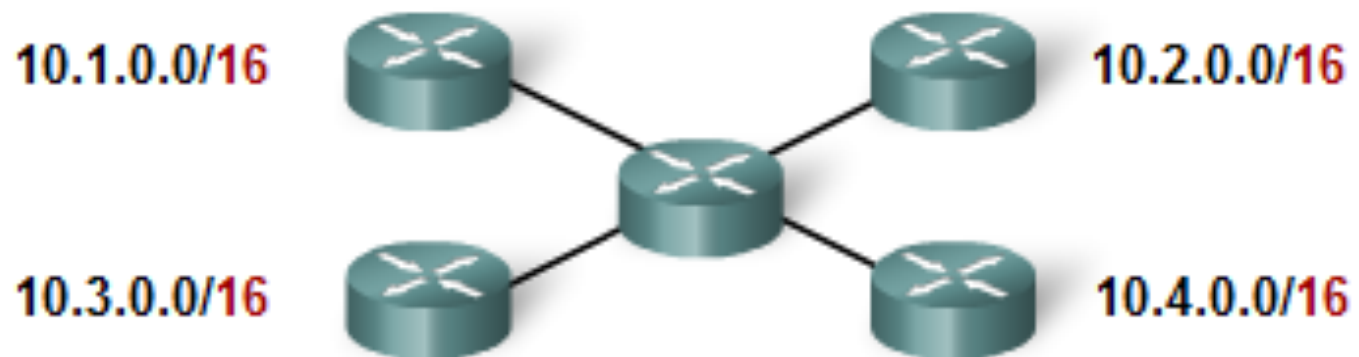
10.2.0.0/16

.

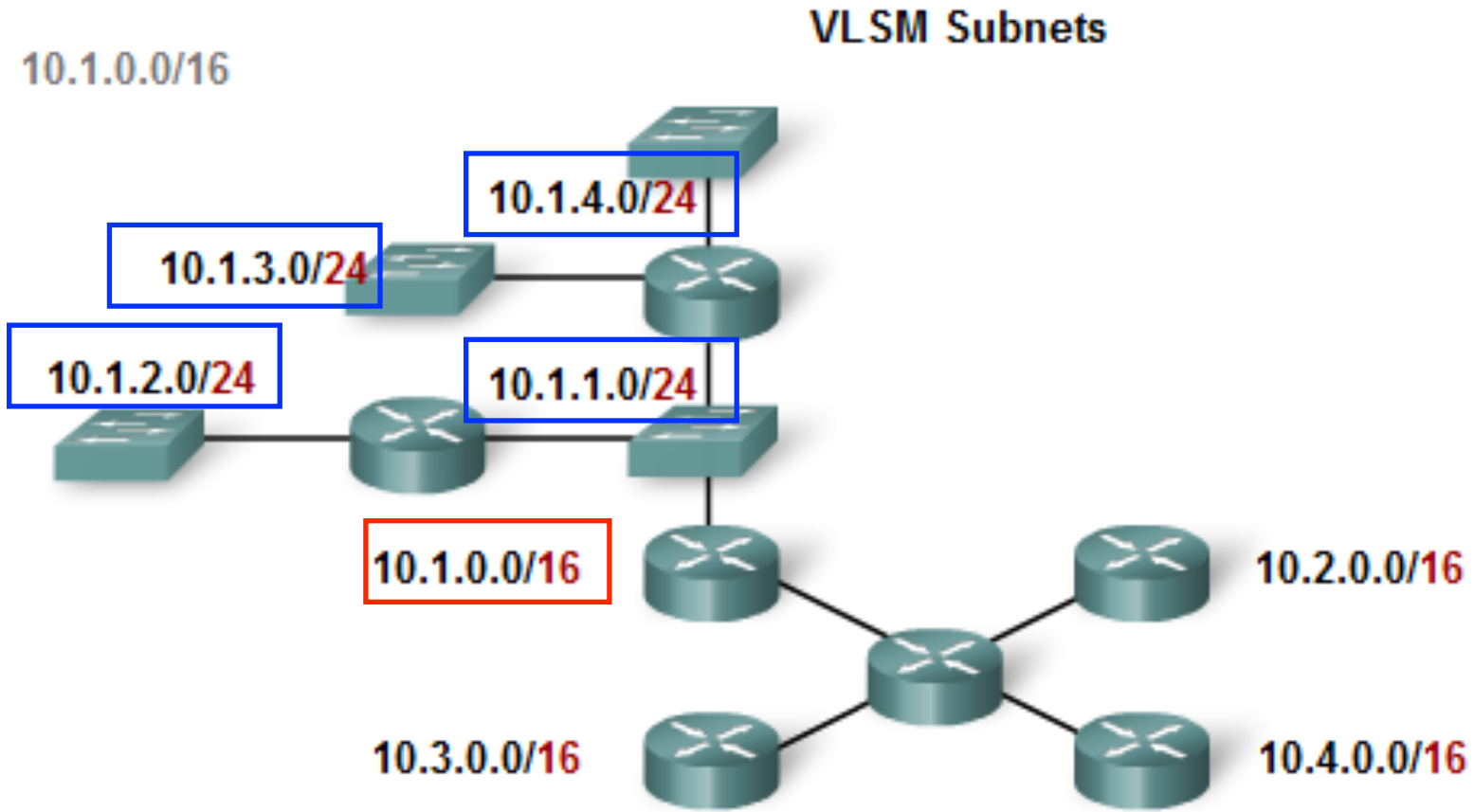
.

.

10.255.0.0/16



# VLSM

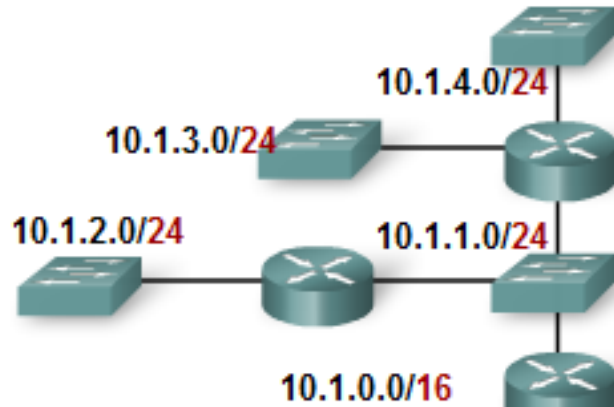


Chacun de ces sous-réseaux /16 peut être aussi subnetté

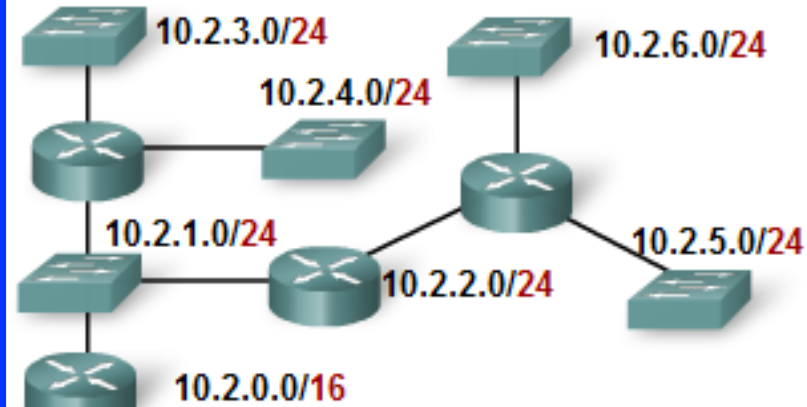
Par exemple, le réseau 10.1.0.0/16 est sous-divisé en plusieurs réseaux /24

### VLSM Subnets

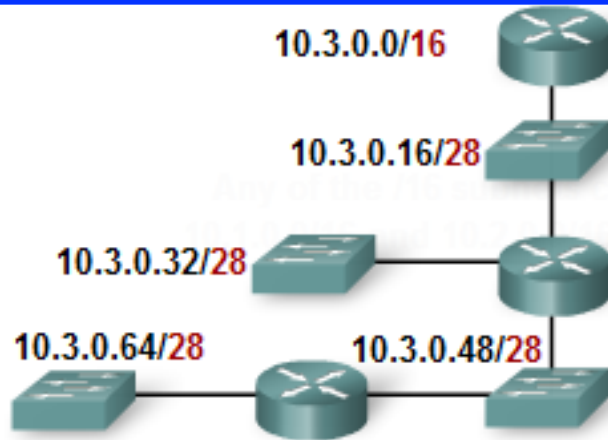
10.1.0.0/16



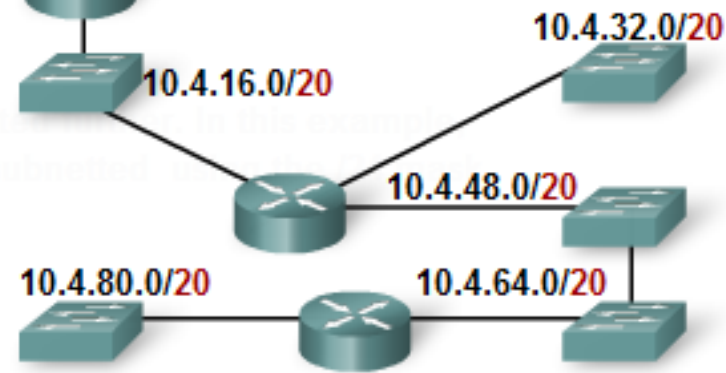
10.2.0.0/16



10.3.0.0/16



10.4.0.0/16

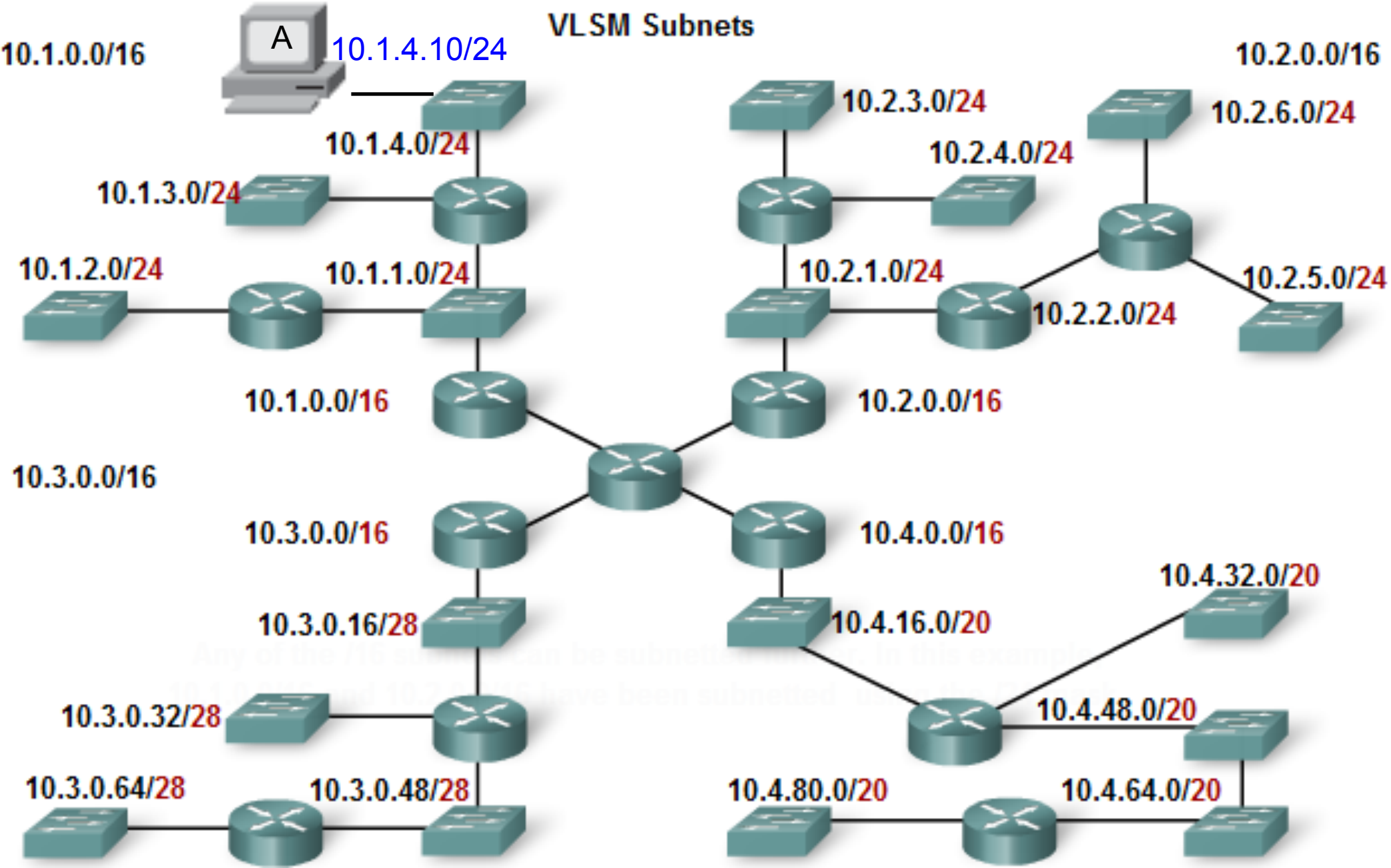


Le réseau 10.1.0.0/16 est divisé en réseaux /24

Le réseau 10.2.0.0/16 est divisé en réseaux /24

Le réseau 10.3.0.0/16 est divisé en réseaux /28

Le réseau 10.4.0.0/16 est divisé en réseaux /20



Les machines ont des adresses dans leur sous-réseau  
 Quelle serait une adresse valide pour la machine A?



# VLSM

Si on sait travailler avec des sous-réseaux, on sait travailler avec le VLSM

Exemple : 10.0.0.0/8

Subdivisé en in /16 sous-réseaux :

10.0.0.0/16

10.1.0.0/16

10.2.0.0/16

10.3.0.0/16

Etc.



Subdivision de l'un des sous-réseaux (10.1.0.0/16)

10.1.0.0/24

10.1.1.0/24

10.1.2.0/24

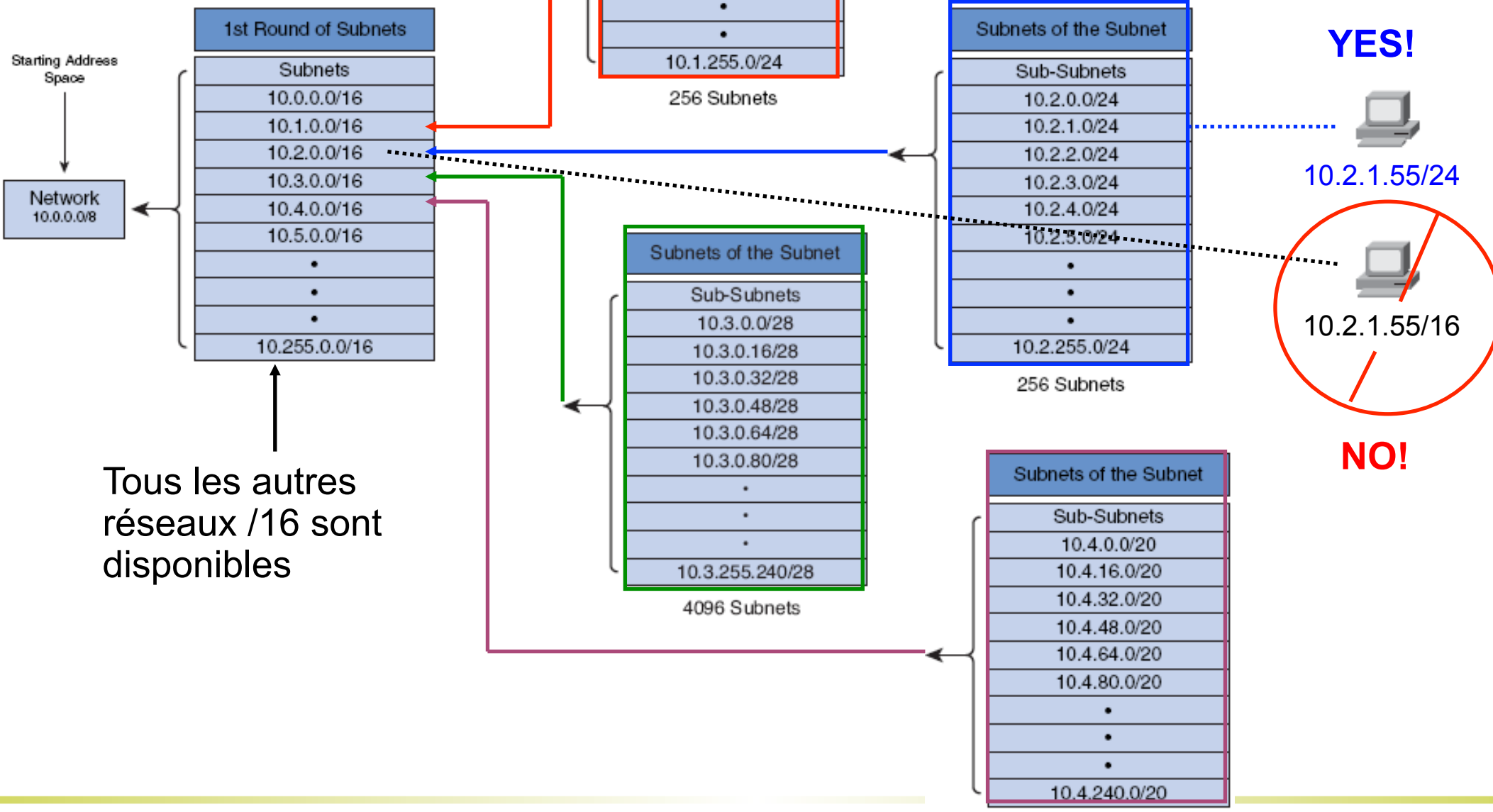
10.1.3.0/24

etc

# VLSM



Une machine devient membre du sous-réseau, elle ne peut plus être membre du réseau "plus grand"



# VLSM – Utilisation d'un tableau

- Un tableau permet d'avoir la vision globale sur les sous-réseaux et leurs utilisations
- On peut faire ceci pour n'importe quel octet de l'adresse
- Pour la simplicité, on représente juste une partie
- Exemple : le réseau 172.16.1.0/24
  - Si nous avons besoin de 5 sous-réseaux ?
  - Quel serait le masque ?
  - Et les adresses de chaque sous-réseau ?
  - Quelle plage d'adresses chaque sous-réseau occuperait ?

Class C Subnet Table

VLSM Chart

Class C Subnet Table	/24 0 (00000000) 0 subnet 254 hosts	/25 .128 (10000000) 0 subnet 128 hosts	/26 .192 (11000000) 2 subnets 62 hosts	/27 .224 (11100000) 4 subnets 30 hosts	/28 .240 (11110000) 16 subnets 14 hosts	/29 .248 (11111000) 32 subnets 6 hosts	/30 .252 (11111100) 62 subnets 2 hosts
0	0	0	0	0	0	0	0 (1-2)
4					(1 - 14)	(1 - 6)	4 (5 - 6)
8						8	8 (9 - 10)
12				(1 - 30)	(8 - 14)	(8 - 14)	12 (13 - 14)
16					.16	.16	16 (17 - 18)
20					(17 - 30)	(17 - 22)	20 (21 - 22)
24						.32	24 (25 - 26)
28						(25 - 30)	28 (29 - 30)
32			(1 - 62)		.32	.32	32 (33 - 34)
36				.32	(33 - 46)	(33 - 38)	36 (37 - 38)
40						.40	40 (41 - 42)
44				(33 - 62)		(41 - 46)	44 (45 - 46)
48					.48	.48	48 (49 - 50)
52					(49 - 54)	(49 - 54)	52 (53 - 54)
56						.56	56 (57 - 58)
60		(1 - 120)			(57 - 62)	(57 - 62)	60 (61 - 62)
64			.64	.64	.64	.64	64 (65 - 66)
68				(65 - 70)	(65 - 70)	(65 - 70)	68 (69 - 70)
72				(65 - 94)	(65 - 78)	.72	72 (73 - 74)
76					(73 - 78)	(73 - 78)	76 (77 - 78)
80					.80	.80	80 (81 - 82)
84					(81 - 86)	(81 - 86)	84 (85 - 86)
88			(65 - 126)		.88	.88	88 (89 - 90)
92					(89 - 94)	(89 - 94)	92 (93 - 94)
96				.96	.96	.96	96 (97 - 98)
100					(97 - 102)	(97 - 102)	100 (101 - 102)
104				(97 - 126)	.104	.104	104 (105 - 106)
108					(105 - 108)	(105 - 108)	108 (107 - 108)
112					.112	.112	112 (113 - 114)
116					(113 - 118)	(113 - 118)	116 (117 - 118)
120					(113 - 126)	.120	120 (121 - 122)
124						(121 - 126)	124 (125 - 126)
128	(1 - 254)	.128	.128	.128	.128	.128	128 (129 - 130)
132					(129 - 142)	(129 - 130)	132 (133 - 134)
136						.136	136 (137 - 138)
140				(129 - 158)		(137 - 142)	140 (141 - 142)
144					.144	.144	144 (145 - 146)
148					(145 - 158)	(145 - 150)	148 (149 - 150)
152						.152	152 (153 - 154)
156			(129 - 191)		(153 - 158)	(153 - 158)	156 (157 - 158)
160				.160	.160	.160	160 (161 - 162)
164					(161 - 174)	(161 - 166)	164 (165 - 166)
168						.168	168 (169 - 170)
172				(161 - 190)		(169 - 174)	172 (173 - 174)
176					.176	.176	176 (177 - 178)
180					(177 - 190)	(177 - 182)	180 (181 - 182)
184						.184	184 (185 - 186)
188					(185 - 190)	(185 - 190)	188 (189 - 190)
192		(129 - 254)	.192	.192	.192	.192	192 (193 - 194)
196					(193 - 206)	(193 - 198)	196 (197 - 198)
200						.200	200 (201 - 202)
204				(193 - 222)		(201 - 206)	204 (205 - 206)
208					.208	.208	208 (209 - 210)
212					(209 - 222)	(209 - 214)	212 (213 - 214)
216						.216	216 (217 - 218)
220					(217 - 222)	(217 - 222)	220 (221 - 222)
224			(191 - 254)	.224	.224	.224	224 (225 - 226)
228					(225 - 238)	(225 - 230)	228 (229 - 230)
232						.232	232 (233 - 234)
236					(233 - 238)	(233 - 238)	236 (237 - 238)
240				(225 - 254)		.240	240 (241 - 242)
244						(241 - 246)	244 (244 - 246)



# VLSM – Utilisation d'un tableau

Réseau : 172.16.1.0/24

Quel serait le masque nécessaire ?

255.255.255.225 (/27)

Et les adresses de chaque sous-réseau ?

- 172.16.1.0/27
- 172.16.1.32/27
- 172.16.1.64/27
- 172.16.1.96/27
- 172.16.1.128/27
- 172.16.1.160/27
- 172.16.1.192/27
- 172.16.1.224/27

Class C Subnet Table	/24 0 subnets/254 hosts (00000000)	/25 0 subnet/126 hosts (10000000)	/26 2 subnets/62 hosts (11000000)	<b>/27 6 subnets/30 hosts (11100000)</b>	/28 14 subnets/14 hosts (11110000)	/29 30 subnets/6 hosts (11111000)	/30 62 subnets/2 hosts (11111100)
0	0	0	0	0	0	0	0 (1-2)
4					(1-4)	(1-6)	4 (5-6)
8					(1-8)	(1-12)	8 (9-16)
12				(1-30)	(1-14)	(1-14)	12 (15-14)
16					.16	.16	16 (17-18)
20					(17-20)	(17-22)	20 (21-22)
24					(17-30)	.24	24 (25-26)
28						(25-30)	28 (29-30)
32			(1-62)	.32	.32	.32	32 (33-34)
36					(33-40)	(33-38)	36 (37-38)
40				(33-62)	.40	.40	40 (41-42)
44					.44	(41-60)	44 (45-46)
48					(40-62)	(40-54)	48 (49-50)
52						.52	52 (53-54)
56						(49-62)	56 (57-58)
60		(1-126)		.64	.64	(57-62)	64 (65-66)
64					(65-70)	.64	68 (69-70)
68					(65-78)	.72	72 (73-74)
72				(65-94)	.80	.80	80 (81-82)
76					(81-84)	(81-84)	84 (85-86)
80					(81-94)	.88	88 (89-90)
84						(89-94)	92 (93-94)
88			(65-126)	.96	.96	.96	96 (97-98)
92					(97-102)	(97-102)	100 (101-102)
96					(97-108)	.104	104 (105-106)
100						(105-108)	108 (107-108)
104				(97-126)	.112	.112	112 (113-114)
108					(113-118)	.120	116 (117-118)
112					(113-126)	(121-126)	120 (121-122)
116						.128	124 (125-126)
120	(1-254)	.128	.128	.128	.128	.128	128 (129-130)
124					(129-142)	.136	132 (133-134)
128				(129-158)	.144	(137-142)	136 (141-142)
132					.144	.144	144 (145-146)
136					(145-150)	(145-149)	148 (149-150)
140					.152	.152	152 (153-154)
144				(129-191)	(153-158)	(153-157)	156 (157-158)
148					.160	.160	160 (161-162)
152					(161-166)	(161-166)	164 (165-166)
156					(161-174)	.168	168 (169-170)
160				(161-190)	(168-174)	(172-173)	172 (173-174)
164					.176	.176	176 (177-178)
168					(177-182)	(177-182)	180 (181-182)
172					.184	.184	184 (185-186)
176					(185-190)	(185-190)	188 (189-190)
180					.192	.192	192 (193-194)
184		(129-254)	.192	.192	(193-198)	(193-197)	196 (197-198)
188					(193-206)	.200	200 (201-202)
192				(193-222)	(201-206)	.204	204 (205-206)
196					.208	.208	208 (209-210)
200					(209-214)	.212	212 (213-214)
204					(209-222)	.216	216 (217-218)
208				(191-254)	(217-222)	.220	220 (221-222)
212					.224	.224	224 (225-226)
216					(225-230)	.228	228 (229-230)
220					(225-238)	.232	232 (233-234)
224				(225-254)	(233-238)	(233-238)	236 (237-238)
228					.240	.240	240 (241-242)
232					(241-246)	(241-246)	244 (244-246)

Quelles seraient les plages d'adresse utilisables ?

- 172.16.1.0/27: 172.16.1.1-172.16.1.31
- 172.16.1.32/27: 172.16.1.33-172.16.1.62
- 172.16.1.64/27: 172.16.1.65-172.16.1.94
- 172.16.1.96/27: 172.16.1.97-172.16.1.126
- Etc.

# VLSM – Utilisation d'un tableau

Et si maintenant on a besoin de 6 sous-réseau /30 pour les liens série ?

Pas de problème, on prend un réseau /27 libre et on le sous-divise

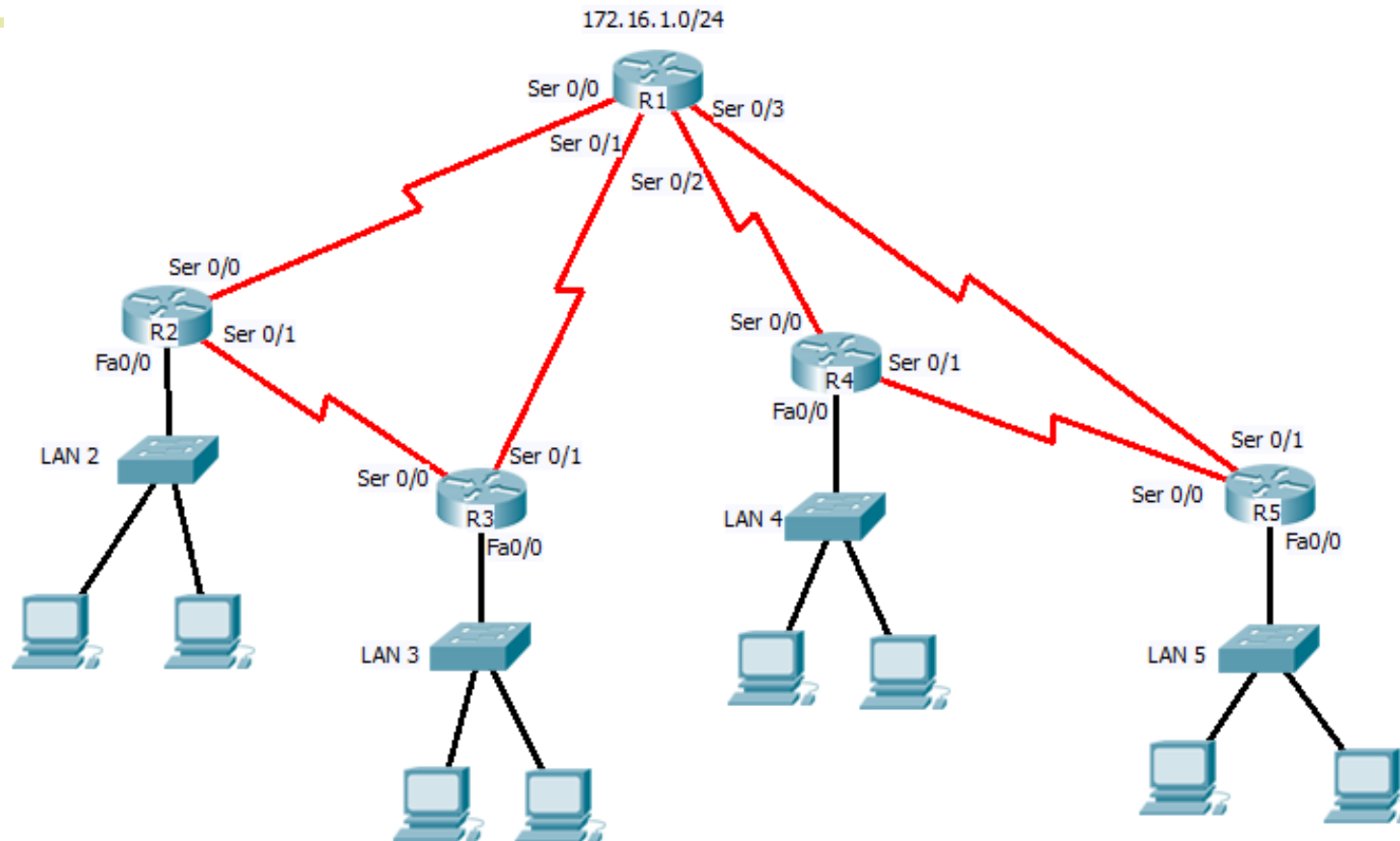
Class C Subnet Table	/24 .0 (00000000) 0 subnets/254 hosts	/25 .128 (10000000) 0 subnets/126 hosts	/26 .192 (11000000) 2 subnets/62 hosts	.224 (11100000) 4 subnets/30 hosts	.240 (11110000) 14 subnets/14 hosts	.252 (11111000) 62 subnets/2 hosts
0	.0	.0	.0	.0	.0	.0 (.1 - .2)
4					(.1 - .14)	.4 (.5 - .6)
8					.8	.8 (.9 - .10)
12					(.9 - .14)	.12 (.13 - .14)
16					.16	.16 (.17 - .18)
20					(.17 - .30)	.20 (.21 - .22)
24					.24	.24 (.25 - .26)
28					(.25 - .30)	.28 (.29 - .30)
32					.32	.32 (.33 - .34)
36					(.33 - .38)	.36 (.37 - .38)
40					.40	.40 (.41 - .42)
44					(.41 - .46)	.44 (.45 - .46)
48					.48	.48 (.49 - .50)
52					(.49 - .62)	.52 (.53 - .54)
56					.56	.56 (.57 - .58)
60					(.57 - .62)	.60 (.61 - .62)
64					.64	.64 (.65 - .66)
68					(.65 - .78)	.68 (.69 - .70)
72					.72	.72 (.73 - .74)
76					(.73 - .78)	.76 (.77 - .78)
80					.80	.80 (.81 - .82)
84					(.81 - .84)	.84 (.85 - .86)
88					.88	.88 (.89 - .90)
92					(.89 - .94)	.92 (.93 - .94)
96					.96	.96 (.97 - .98)
100					(.97 - .108)	.100 (.101 - .102)
104					.104	.104 (.105 - .106)
108					(.105 - .108)	.108 (.109 - .110)
112					.112	.112 (.113 - .114)
116					(.113 - .126)	.116 (.117 - .118)
120					.120	.120 (.121 - .122)
124					(.121 - .126)	.124 (.125 - .126)
128					.128	.128 (.129 - .130)
132					(.129 - .191)	.132 (.133 - .134)
136					.136	.136 (.137 - .138)
140					(.137 - .190)	.140 (.141 - .142)
144					.144	.144 (.145 - .146)
148					(.145 - .198)	.148 (.149 - .150)
152					.152	.152 (.153 - .154)
156					(.153 - .191)	.156 (.157 - .158)
160					.160	.160 (.161 - .162)
164					(.161 - .190)	.164 (.165 - .166)
168					.168	.168 (.169 - .170)
172					(.169 - .190)	.172 (.173 - .174)
176					.176	.176 (.177 - .178)
180					(.177 - .190)	.180 (.181 - .182)
184					.184	.184 (.185 - .186)
188					(.185 - .190)	.188 (.189 - .190)
192					.192	.192 (.193 - .194)
196					(.193 - .206)	.196 (.197 - .198)
200					.200	.200 (.201 - .202)
204					(.201 - .206)	.204 (.205 - .206)
208					.208	.208 (.209 - .210)
212					(.209 - .214)	.212 (.213 - .214)
216					.216	.216 (.217 - .218)
220					(.217 - .222)	.220 (.221 - .222)
224					.224	.224 (.225 - .226)
228					(.225 - .238)	.228 (.229 - .230)
232					.232	.232 (.233 - .234)
236					(.233 - .238)	.236 (.237 - .238)
240					.240	.240 (.241 - .242)
244					(.241 - .246)	.244 (.244 - .246)

/26 .192 (11000000) 2 subnets 62 hosts	/27 .224 (11100000) 6 subnets 30 hosts	/28 .240 (11110000) 14 subnets 14 hosts	/29 .248 (11111000) 30 subnets 6 hosts	/30 .252 (11111100) 62 subnets 2 hosts
.0	.0	.0	.0	.0 (.1 - .2)
		(.1 - .14)	(.1 - .6)	.4 (.5 - .6)
	(.1 - .30)	.8	.8	.8 (.9 - .10)
		(.9 - .14)	.12	.12 (.13 - .14)
		.16	.16	.16 (.17 - .18)
		(.17 - .30)	.20	.20 (.21 - .22)
		.24	.24	.24 (.25 - .26)
		(.25 - .30)	.28	.28 (.29 - .30)
(.1 - .62)	.32	.32	.32	.32 (.33 - .34)
	(.33 - .62)	(.33 - .46)	.40	.36 (.37 - .38)
		.40	.40	.40 (.41 - .42)
		(.41 - .46)	.44	.44 (.45 - .46)
		.48	.48	.48 (.49 - .50)
		(.49 - .62)	.52	.52 (.53 - .54)
		.56	.56	.56 (.57 - .58)
		(.57 - .62)	.60	.60 (.61 - .62)

Il nous restent encore 7 réseaux /27

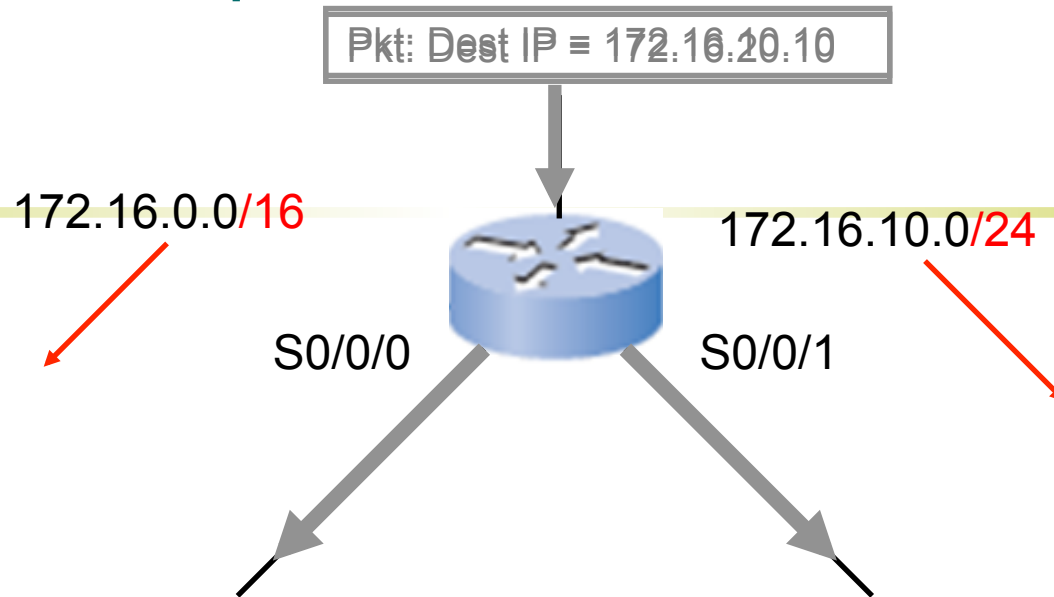
8 sous-réseaux /30

# Application sur la topologie



Grâce au tableau, maintenant on sait quels plages d'adresse sont destinées à chaque sous-réseau, ainsi que les plages encore disponibles

# Priorité des masques



Est-ce que deux routes pour une même destination (spécifique et agrégée) peuvent coexister dans un routeur ?

Si un paquet est destiné à **172.16.10.10**, quel chemin il prendra ?

Le paquet utilise la règle plus précise (masque plus long) **172.16.10.0/24**, ainsi **S0/0/1** sera utilisé pour relayer le paquet

Et si un paquet est destiné à **172.16.20.10** ?

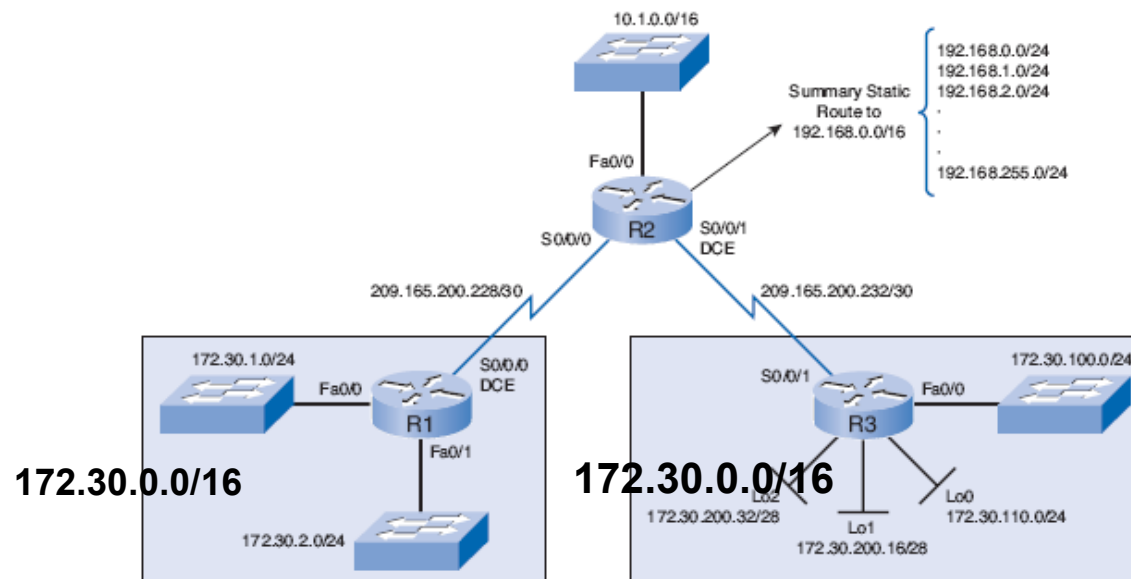
Dans ce cas, la règle spécifique ne correspond pas (les 24 premiers bits ne sont pas identiques), alors la règle **172.16.0.0/16** sera utilisée pour relayer le paquet à travers **S0/0/0**

Uniquement si les 16 premiers bits correspondent !

# LE PROTOCOLE RIPV2

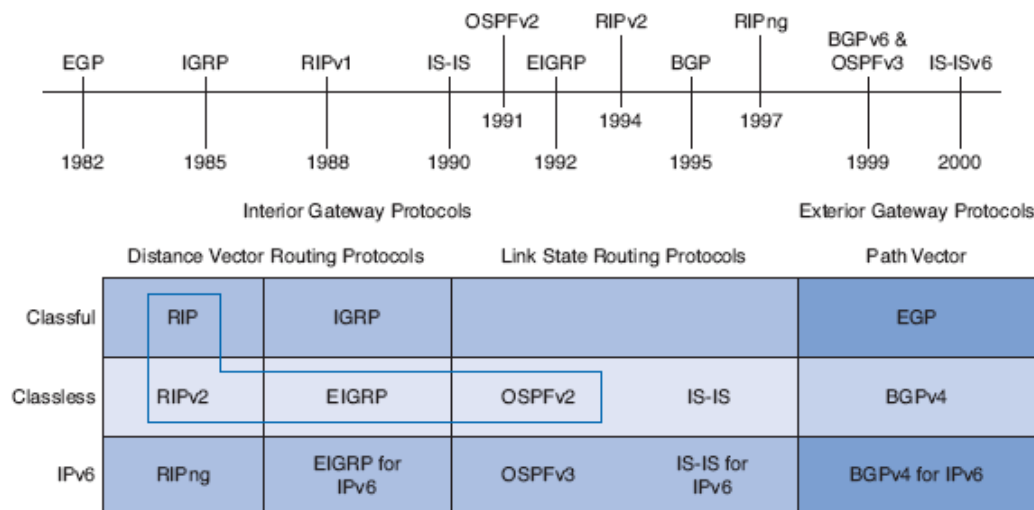
# Les Limitations de RIPv1

- Les réseaux non-contigus, où une adresse classful comme 172.30.0.0/16 est séparé par un ou plusieurs réseaux.
- 172.30.0.0/16 est séparé par les réseaux :
  - 209.165.200.228/30
  - 209.165.200.232/30
- Les protocoles classful n'incluent pas les informations de routage nécessaires aux réseaux non-contigus.



# RIPv1 et RIPv2

- RIPv2 est une extension de RIPv1, pas un nouveau protocole.
  - **Les adresses des next-hop** incluses dans la table de routage
  - Utilisation de **multicast** pour l'envoi des updates
  - Possibilité d'**authentification**
- Les deux versions de RIP partagent ces caractéristiques et limitations:
  - Utilisation de **hold-down timers** pour éviter les boucles
  - Utilisation de **split horizon** et **split horizon with poison reverse** aussi pour éviter les boucles
  - Utilisation de **triggered updates** lors des modifications pour accélérer la convergence
  - **Maximum hop count de 15 sauts**



Highlighted routing protocols are the focus of this course.

# Configuration de RIPv2

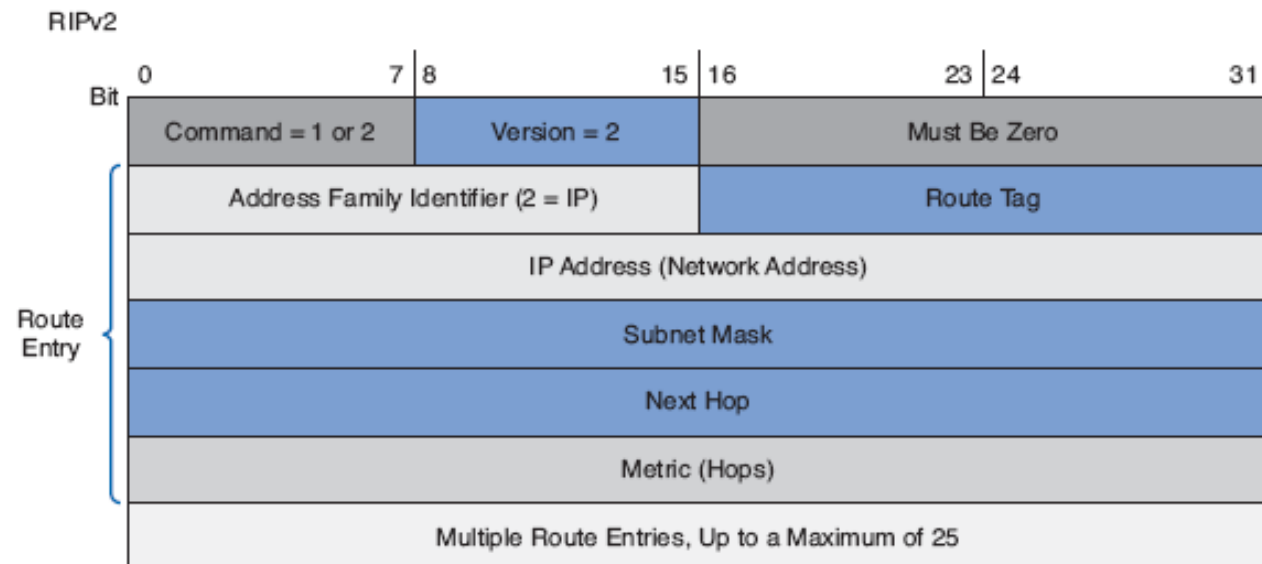
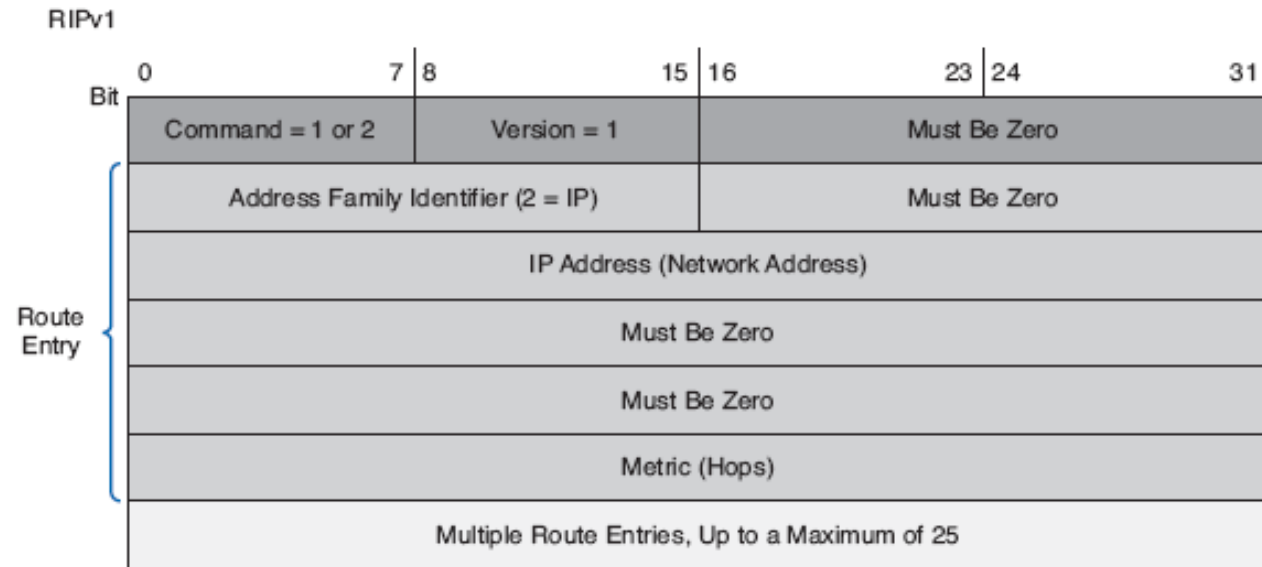
---

- Activation et vérification de RIPv2
- Agrégation automatique et RIPv2
- Désactivation de l'agrégation automatique dans RIPv2
- Vérification des updates RIPv2



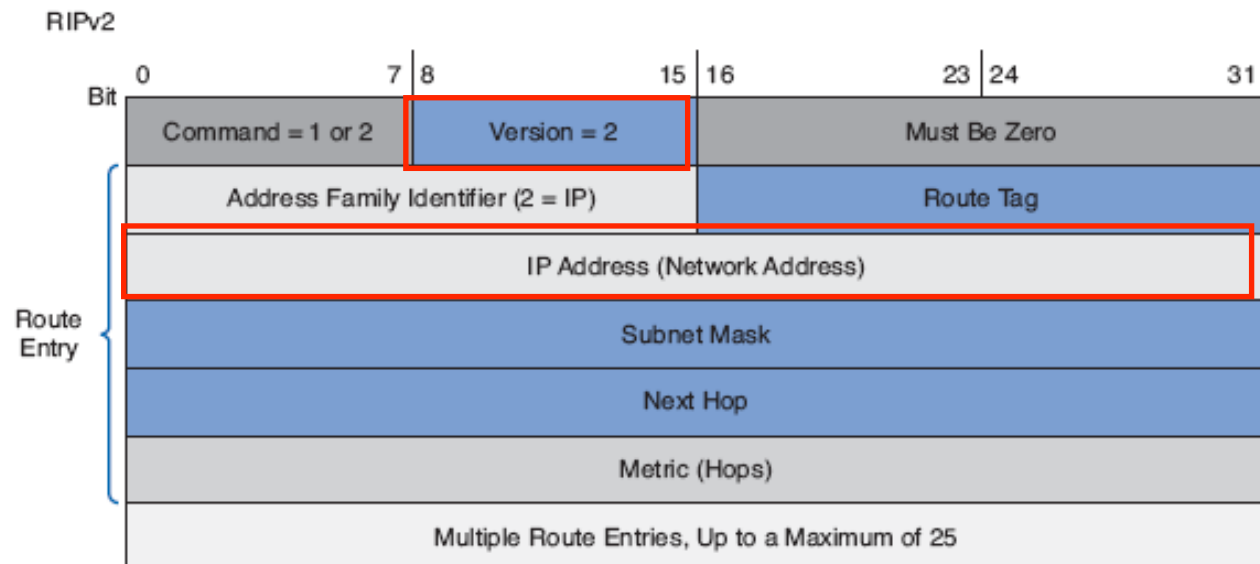
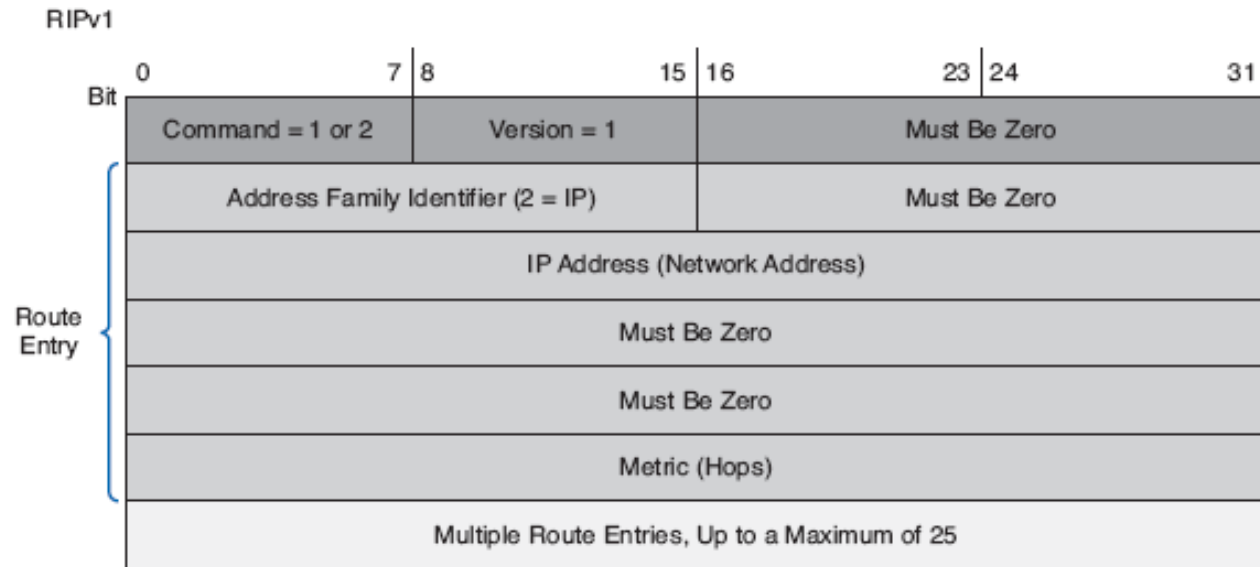
# Configuration de RIPv2

- La configuration de RIPv2 est similaire à celle de RIPv1, il suffit d'indiquer la commande **version 2**.
- Si les commandes de base sont identiques à RIPv1, **le résultat avec RIPv2 est différent**, car VLSM et les réseaux non-contigus sont possibles désormais.



# Activation et Vérification de RIPv2

- RIPv2 est défini dans le **RFC 1723**.
- Le format des messages RIPv2 inclut les masques dans les updates.
- Le résultat est que le masque n'est plus dépendant de l'interface d'entrée ni du grand réseau classfull



# Activation et Vérification de RIPv2

- Par défaut RIPv1 est activé
  - Le routeur n'envoie que des messages RIPv1, même s'il peut traiter des messages RIPv1 et RIPv2.
  - Il suffit d'ignorer les champs RIPv2.
- RIPv2, par contre, ignore les messages RIPv1.
- **INFO** : Les commandes d'interface `ip rip send` et `ip rip receive` peuvent être utilisés pour forcer la compatibilité entre les versions.

```
R2# show ip protocols
<output omitted>
  Default version control: send version 1, receive any version
  Interface                Send      Recv     Triggered RIP Key-chain
  Serial0/0/0              1       1 2
  Serial0/0/1              1       1 2
  Automatic network summarization is in effect
<output omitted >
```

# Activation et Vérification de RIPv2

- La commande `version 2` est utilisée pour activer la Version 2.
- Cette commande doit être entrée sur toutes les machines du domaine.

```
R1 (config) # router rip  
R1 (config-router) # version 2
```

```
R2 (config) # router rip  
R2 (config-router) # version 2
```

```
R3 (config) # router rip  
R3 (config-router) # version 2
```

# Activation et Vérification de RIPv2

```
R2# show ip protocols
```

```
Routing Protocol is "rip"
```

```
  Sending updates every 30 seconds, next due in 1 seconds
```

```
  Invalid after 180 seconds, hold down 180, flushed after 240
```

```
  Outgoing update filter list for all interfaces is
```

```
  Incoming update filter list for all interfaces is
```

```
  Redistributing: static, rip
```

```
  Default version control: send version 2, receive version 2
```

Interface	Send	Recv	Triggered RIP Key-chain
Serial0/0/0	2	2	
Serial0/0/1	2	2	

```
  Automatic network summarization is in effect
```

```
<output omitted for brevity>
```

# Restauration de RIP Version 1

- Le comportement de RIPv1 peut être rétabli en utilisant ces deux commandes (différentes en ce qui concerne l'envoi et la réception des messages) :
  - **version 1**
  - **no version**
- Ces commandes doivent être entrées sur toutes les machines du domaine.

```
R1(config)# router rip
R1(config-router)# version 1

!or

R1(config)# router rip
R1(config-router)# no version
```

# Agrégation automatique et RIPv2

- On vient d'activer RIPv2 cependant...
- La commande `show ip route` montre toujours une route agrégée **172.30.0.0/16** avec les **mêmes deux chemins de coût identique**.

```
R2# show ip route
```

```
R    172.30.0.0/16 [120/1] via 209.165.200.230, 00:00:28, Serial0/0/0
      [120/1] via 209.165.200.234, 00:00:18, Serial0/0/1
    209.165.200.0/30 is subnetted, 2 subnets
C      209.165.200.232 is directly connected, Serial0/0/1
C      209.165.200.228 is directly connected, Serial0/0/0
    10.0.0.0/16 is subnetted, 1 subnets
C      10.1.0.0 is directly connected, FastEthernet0/0
S     192.168.0.0/16 is directly connected, Null0
```

# Agrégation automatique et RIPv2

- Les routeurs **R1 et R3 n'incluent toujours pas les sous-réseaux de 172.30.0.0.**
- La **seule différence pour l'instant** est que R1 et R3 ont chacun une route vers **192.168.0.0/16**.
  - C'est une route statique (**CIDR**) configurée en R2 et redistribuée par RIP.
- Quel est le problème ?

```
R1# show ip route
```

```
172.30.0.0/24 is subnetted, 2 subnets
```

```
C    172.30.2.0 is directly connected, Loopback0
```

```
C    172.30.1.0 is directly connected, FastEthernet0/0
```

```
209.165.200.0/30 is subnetted, 2 subnets
```

```
R    209.165.200.232 [120/1] via 209.165.200.229, 00:00:04, Serial0/0/0
```

```
C    209.165.200.228 is directly connected, Serial0/0/0
```

```
R    10.0.0.0/8 [120/1] via 209.165.200.229, 00:00:04, Serial0/0/0
```

```
R    192.168.0.0/16 [120/1] via 209.165.200.229, 00:00:04, Serial0/0/0
```



# Agrégation automatique et RIPv2

- Avec le debug, on voit que **RIPv2 envoie les réseaux avec leurs masques.**
- Cependant, la **route envoyée est la route agrégée, 172.30.0.0/16**
  - pourquoi 172.30.1.0/24 et 172.30.2.0/24 ne sont pas envoyés ?

```
R1# debug ip rip
RIP: sending v2 update to 224.0.0.9 via Serial0/0/0 (209.165.200.230) ↑
RIP: build update entries
      172.30.0.0/16 via 0.0.0.0, metric 1, tag 0
<output omitted for brevity>
RIP: received v2 update from 209.165.200.229 on Serial0/0/0
      10.0.0.0/8 via 0.0.0.0 in 1 hops
      192.168.0.0/16 via 0.0.0.0 in 1 hops
      209.165.200.232/30 via 0.0.0.0 in 1 hops
```

# Agrégation automatique et RIPv2

- ***Par défaut, RIPv2 fait l'agrégation automatique lors du passage entre deux réseaux classful, exactement comme RIPv1.***
- R1 et R3 agrègent encore leurs sous-réseaux 172.30.0.0 lorsqu'ils envoient les updates vers les interfaces sur les réseaux 209.165.200.228 et 209.165.200.232.

```
R1# show ip protocols
Routing Protocol is "rip"
<output omitted>
  Default version control: send version 2, receive version 2
  Interface          Send    Recv    Triggered RIP Key-chain
  FastEthernet0/0    2      2
  FastEthernet0/1    2      2
  Serial0/1/0        2      2
```

**Automatic network summarization is in effect**

# Désactivation de l'agrégation automatique RIPv2

- Pour modifier le comportement de RIPv2, il faut utiliser la commande **no auto-summary**

```
R2(config)# router rip
R2(config-router)# no auto-summary
```

```
R3(config)# router rip
R3(config-router)# no auto-summary
```

```
R1(config)# router rip
R1(config-router)# no auto-summary
```

```
R1# show ip protocols
<output omitted>
  Automatic network summarization is not in effect
<output omitted>
```

# Vérification des updates RIPv2

- La table de routage de **R2** contient maintenant tous les sous-réseaux de **172.30.0.0/16**.
- La route agrégée avec deux chemins de même coût n'existe plus.
- Chaque sous-réseau et masque ont des entrées spécifiques, ainsi que des interfaces de sortie et passerelles individuelles.

```
R2# show ip route
```

```
172.30.0.0/16 is variably subnetted, 6 subnets, 2 masks
```

```
R 172.30.200.32/28 [120/1] via 209.165.200.234, 00:00:09, Serial0/0/1
```

```
R 172.30.200.16/28 [120/1] via 209.165.200.234, 00:00:09, Serial0/0/1
```

```
R 172.30.2.0/24 [120/1] via 209.165.200.230, 00:00:03, Serial0/0/0
```

```
R 172.30.1.0/24 [120/1] via 209.165.200.230, 00:00:03, Serial0/0/0
```

```
R 172.30.100.0/24 [120/1] via 209.165.200.234, 00:00:09, Serial0/0/1
```

```
R 172.30.110.0/24 [120/1] via 209.165.200.234, 00:00:09, Serial0/0/1
```

```
209.165.200.0/30 is subnetted, 2 subnets
```

```
C 209.165.200.232 is directly connected, Serial0/0/1
```

```
C 209.165.200.228 is directly connected, Serial0/0/0
```

```
10.0.0.0/16 is subnetted, 1 subnets
```

```
C 10.1.0.0 is directly connected, FastEthernet0/0
```

```
S 192.168.0.0/16 is directly connected, Null0
```

# Vérification des updates RIPv2

- Ceci permet la convergence totale des tables de routage

```
R1# show ip route
 172.30.0.0/16 is variably subnetted, 6 subnets, 2 masks
R   172.30.200.32/28 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
R   172.30.200.16/28 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
C   172.30.2.0/24 is directly connected, Loopback0
C   172.30.1.0/24 is directly connected, FastEthernet0/0
R   172.30.100.0/24 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
R   172.30.110.0/24 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
 209.165.200.0/30 is subnetted, 2 subnets
R   209.165.200.232 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
C   209.165.200.228 is directly connected, Serial0/0/0
 10.0.0.0/16 is subnetted, 1 subnets
R   10.1.0.0 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
R   192.168.0.0/16 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
```

# Vérification des updates RIPv2

- Ceci permet la convergence totale des tables de routage

```
R3# show ip route
 172.30.0.0/16 is variably subnetted, 6 subnets, 2 masks
C   172.30.200.32/28 is directly connected, Loopback2
C   172.30.200.16/28 is directly connected, Loopback1
R   172.30.2.0/24 [120/2] via 209.165.200.233, 00:00:01, Serial0/0/1
R   172.30.1.0/24 [120/2] via 209.165.200.233, 00:00:01, Serial0/0/1
C   172.30.100.0/24 is directly connected, FastEthernet0/0
C   172.30.110.0/24 is directly connected, Loopback0
 209.165.200.0/30 is subnetted, 2 subnets
C   209.165.200.232 is directly connected, Serial0/0/1
R   209.165.200.228 [120/1] via 209.165.200.233, 00:00:02, Serial0/0/1
 10.0.0.0/16 is subnetted, 1 subnets
R   10.1.0.0 [120/1] via 209.165.200.233, 00:00:02, Serial0/0/1
R  192.168.0.0/16 [120/1] via 209.165.200.233, 00:00:02, Serial0/0/1
```

# Vérification des updates RIPv2

- Les messages envoyés/reçus ont des entrées pour chaque sous-réseau, au lieu d'une seule adresse agrégée classful.

```
R2# debug ip rip
RIP: received v2 update from 209.165.200.234 on Serial0/0/1
    172.30.100.0/24 via 0.0.0.0 in 1 hops
    172.30.110.0/24 via 0.0.0.0 in 1 hops
    172.30.200.16/28 via 0.0.0.0 in 1 hops
    172.30.200.32/28 via 0.0.0.0 in 1 hops
RIP: sending v2 update to 224.0.0.9 via Serial0/0/0 (209.165.200.229) ↑
RIP: build update entries
    10.1.0.0/16 via 0.0.0.0, metric 1, tag 0
    172.30.100.0/24 via 0.0.0.0, metric 2, tag 0
    172.30.110.0/24 via 0.0.0.0, metric 2, tag 0
    172.30.200.16/28 via 0.0.0.0, metric 2, tag 0
    172.30.200.32/28 via 0.0.0.0, metric 2, tag 0
    192.168.0.0/16 via 0.0.0.0, metric 1, tag 0
    209.165.200.232/30 via 0.0.0.0, metric 1, tag 0
```

# Vérification des updates RIPv2

- On peut observer aussi que les **updates utilisent maintenant l'adresse de multicast 224.0.0.9.**
- **RIPv1** envoyait des updates à l'adresse **broadcast 255.255.255.255.**
- D'habitude, les updates par multicast :
  - Consomment moins bande passante du réseau.
  - Surchargent moins la CPU des dispositifs qui n'ont pas RIP activé.

```
R2# debug ip rip  
RIP: sending v2 update to 224.0.0.9 via Serial0/0/0 (209.165.200.229) ↑
```



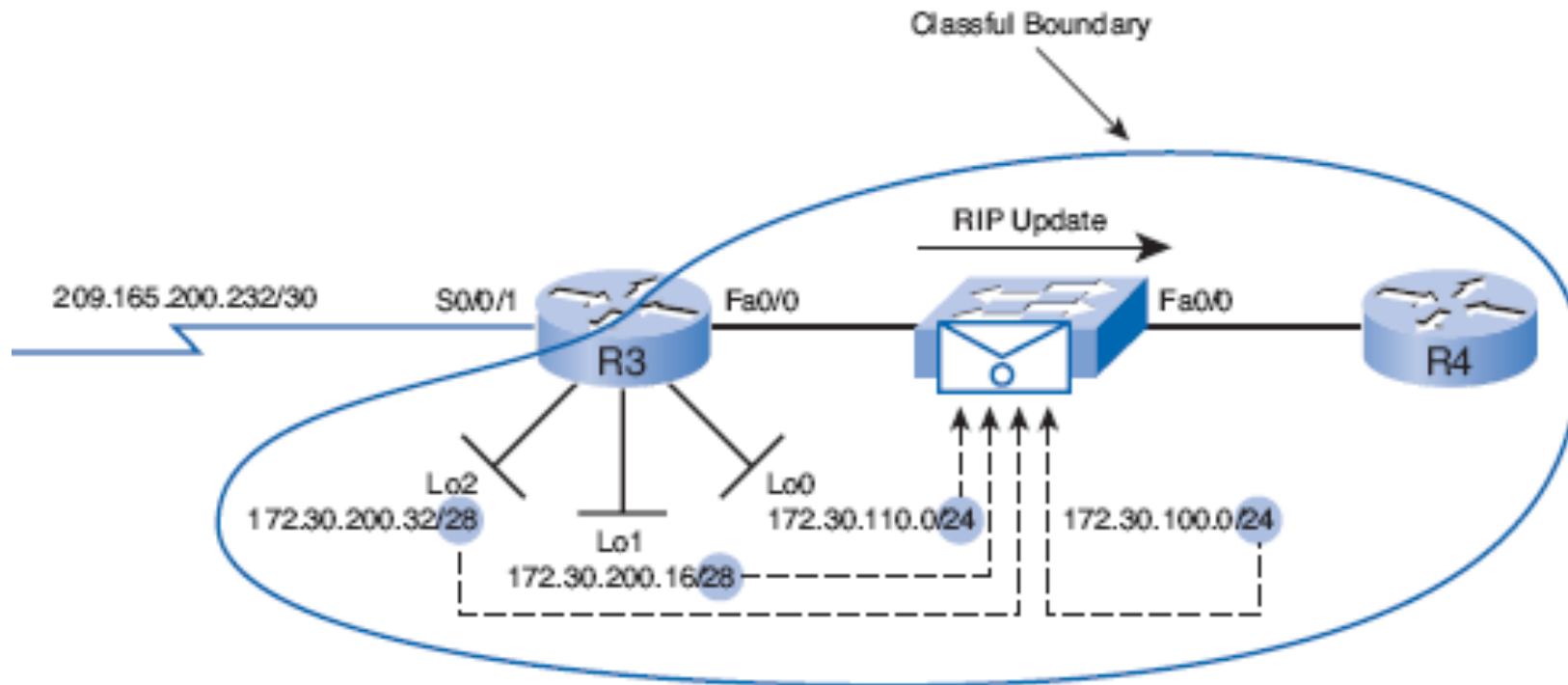
# VLSM et CIDR

---

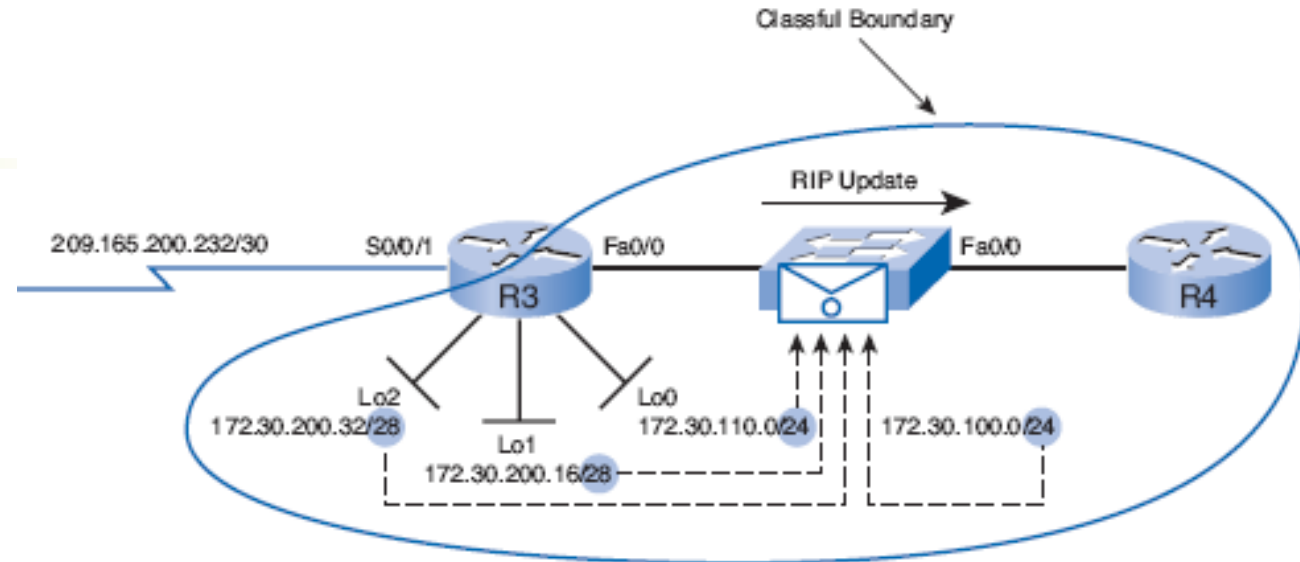
- RIPv2 et VLSM
- RIPv2 et CIDR

# RIPv2 et VLSM

- Avec RIPv2, R3 peut maintenant envoyer tous les sous-réseaux 172.30.0.0 dans les updates vers R4
- Ceci est possible car RIPv2 inclut les masques dans les update.



# RIPv2 et VLSM



```
R3# debug ip rip
```

```
RIP: sending v2 update to 224.0.0.9 via FastEthernet0/0 (172.30.100.1) ^
```

```
RIP: build update entries
```

```
10.1.0.0/16 via 0.0.0.0, metric 2, tag 0
```

```
172.30.1.0/24 via 0.0.0.0, metric 3, tag 0
```

```
172.30.2.0/24 via 0.0.0.0, metric 3, tag 0
```

```
172.30.110.0/24 via 0.0.0.0, metric 1, tag 0
```

```
172.30.200.16/28 via 0.0.0.0, metric 1, tag 0
```

```
172.30.200.32/28 via 0.0.0.0, metric 1, tag 0
```

```
192.168.0.0/16 via 0.0.0.0, metric 2, tag 0
```

```
209.165.200.228/30 via 0.0.0.0, metric 2, tag 0
```

```
209.165.200.232/30 via 0.0.0.0, metric 1, tag 0
```

- *Les Supernets ont des masques plus petits que les masques classful (/16 ici, au lieu de /24).*
- Pour qu'un supernet soit inclus dans les updates, le **protocole de routage doit pouvoir transporter ce masque**
- C'est à dire, il faut un protocole de routage **classless**, comme RIPv2.

```
R2(config)# ip route 192.168.0.0 255.255.0.0 Null0
```

- **Le supernet CIDR est inclus dans l'update** envoyé par R2.
- **Ceci est indépendant de l'agrégation automatique** dans RIPv2, qui ne nécessite pas d'être désactivée pour inclure les supernets dans les updates.

```
R2# debug ip rip
RIP: sending v2 update to 224.0.0.9 via Serial0/0/0 (209.165.200.229) ↑
RIP: build update entries
    10.1.0.0/16 via 0.0.0.0, metric 1, tag 0
    172.30.100.0/24 via 0.0.0.0, metric 2, tag 0
    172.30.110.0/24 via 0.0.0.0, metric 2, tag 0
    172.30.200.16/28 via 0.0.0.0, metric 2, tag 0
    172.30.200.32/28 via 0.0.0.0, metric 2, tag 0
    192.168.0.0/16 via 0.0.0.0, metric 1, tag 0
    209.165.200.232/30 via 0.0.0.0, metric 1, tag 0
```

# RIPv2 et CIDR

- La table de routage de R1 indique qu'il a reçu un supernet de R2
- R1 applique le masque qui a été envoyé avec l'update.

```
R1# show ip route
 172.30.0.0/16 is variably subnetted, 6 subnets, 2 masks
R   172.30.200.32/28 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
R   172.30.200.16/28 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
C   172.30.2.0/24 is directly connected, Loopback0
C   172.30.1.0/24 is directly connected, FastEthernet0/0
R   172.30.100.0/24 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
R   172.30.110.0/24 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
 209.165.200.0/30 is subnetted, 2 subnets
R   209.165.200.232 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
C   209.165.200.228 is directly connected, Serial0/0/0
 10.0.0.0/16 is subnetted, 1 subnets
R   10.1.0.0 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
R   192.168.0.0/16 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
```

# Vérification et Dépannage de RIPv2

---

- Vérification et Dépannage
- Problèmes communs avec RIPv2

# La commande `show ip route`

- **LA** commande pour vérifier la **convergence du réseau**.
- Il faut chercher les routes qui **doivent** être dans la table ainsi que celles qui **ne doivent pas** être présentes

```
R1# show ip route
 172.30.0.0/16 is variably subnetted, 6 subnets, 2 masks
R   172.30.200.32/28 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
R   172.30.200.16/28 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
C   172.30.2.0/24 is directly connected, Loopback0
C   172.30.1.0/24 is directly connected, FastEthernet0/0
R   172.30.100.0/24 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
R   172.30.110.0/24 [120/2] via 209.165.200.229, 00:00:01, Serial0/0/0
 209.165.200.0/30 is subnetted, 2 subnets
R   209.165.200.232 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
C   209.165.200.228 is directly connected, Serial0/0/0
 10.0.0.0/16 is subnetted, 1 subnets
R   10.1.0.0 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
R   192.168.0.0/16 [120/1] via 209.165.200.229, 00:00:02, Serial0/0/0
```



# La commande `show ip interface brief`

- Si un **réseau n'est pas présent** dans la table de routage, peut-être c'est juste son **interface qui est down ou mal configurée**.
- La commande `show ip interface brief` vérifie rapidement l'état des interfaces.

```
R1# show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	172.30.1.1	YES	NVRAM	up	up
FastEthernet0/1	172.30.2.1	YES	NVRAM	up	up
Serial0/0/0	209.165.200.230	YES	NVRAM	up	up
Serial0/0/1	unassigned	YES	NVRAM	down	down

# show ip protocols

- **show ip protocols** vérifie plusieurs détails tels que l'état de RIP, la version, l'agrégation automatique, et les réseaux inclus avec la commande **network**.

```
R1# show ip protocols
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 29 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface          Send    Recv    Triggered RIP Key-chain
  FastEthernet0/0      2       2
  FastEthernet0/1      2       2
  Serial0/0/0          2       2
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    172.30.0.0
    209.165.200.0
  Routing Information Sources:
    Gateway             Distance    Last Update
  209.165.200.229      120        00:00:18
  Distance: (default is 120)↑
```

# debug ip rip

- **debug ip rip** permet d'examiner le contenu des updates envoyés et reçus.
- Parfois **une route est reçue mais n'est pas rajoutée à la table de routage.**
- Une raison possible est qu'une **route statique est configurée pour le même réseau.**

```
R2# debug ip rip
RIP: received v2 update from 209.165.200.234 on Serial0/0/1
  172.30.100.0/24 via 0.0.0.0 in 1 hops
  172.30.110.0/24 via 0.0.0.0 in 1 hops
  172.30.200.16/28 via 0.0.0.0 in 1 hops
  172.30.200.32/28 via 0.0.0.0 in 1 hops
RIP: sending v2 update to 224.0.0.9 via Serial0/0/0 (209.165.200.229)†
RIP: build update entries
  10.1.0.0/16 via 0.0.0.0, metric 1, tag 0
  172.30.100.0/24 via 0.0.0.0, metric 2, tag 0
  172.30.110.0/24 via 0.0.0.0, metric 2, tag 0
  172.30.200.16/28 via 0.0.0.0, metric 2, tag 0
  172.30.200.32/28 via 0.0.0.0, metric 2, tag 0
  192.168.0.0/16 via 0.0.0.0, metric 1, tag 0
  209.165.200.232/30 via 0.0.0.0, metric 1, tag 0
```

- Avec **ping** on teste la connectivité aller-retour entre les réseaux

```
R2# ping 172.30.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.2.1, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
R2# ping 172.30.100.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.100.1, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/28 ms
```

# show running-config

```
R1# show running-config
!
hostname R1
!
interface FastEthernet0/0
 ip address 172.30.1.1 255.255.255.0
!
interface FastEthernet0/1
 ip address 172.30.2.1 255.255.255.0
!
interface Serial0/0/0
 ip address 209.165.200.230 255.255.255.252
 clock rate 64000
!
router rip
 version 2
 network 172.30.0.0
 network 209.165.200.0
 no auto-summary
!
<some output omitted for brevity>
```

# Problèmes communs avec RIPv2

- **Version :**
  - Même si RIPv1 et RIPv2 peuvent être forcés à devenir compatibles, RIPv1 ne supporte pas les réseaux non-contigus, VLSM, ou supernets CIDR.
- **Les entrées `network` :**
  - Une entrée **`network` incorrectement configurée ou manquante** est une source fréquente de problèmes.
  - La commande **`network`** a deux fonctions :
    - Activer le protocole de routage pour que celui-ci envoie et reçoit des updates sur toutes les interfaces dans ce réseau.
    - Rajouter le réseau configuré dans les updates envoyés aux voisins.
  - Une entrée **`network`** incorrecte ou manquante empêcherait les updates d'être envoyés ou reçus par une interface.
- **Agrégation automatique :**
  - Si on attend des updates contenant les sous-réseaux, il faut s'assurer que l'agrégation automatique est désactivée avec la commande **`no auto-summary`**.

# LE PROTOCOLE EIGRP

# EIGRP ?

- Les protocoles de routage à vecteur distance ont plus de difficulté à converger que les algorithmes de routage à états de liens
  - EIGRP s'est inspiré de quelques techniques (mais ça reste du V.D.)
  - Parfois considéré comme un protocole mixte (ce qui est faux)
- EIGRP est une version avancée d'IGRP (un vieux protocole CISCO ne plus utilisé)
  - Converge plus vite qu'IGRP
  - Tous 2 propriétaires Cisco
  - EIGRP envoie d'abord toutes ses informations de routage à un voisin et ensuite seulement des mises à jour
    - IGRP envoie régulièrement (toutes les 90 s.) la totalité de sa table de routage
  - EIGRP fonctionne avec Novell IPX et Apple AppleTalk, en plus d'IP, contrairement à IGRP



# EIGRP

	Interior Gateway Protocols		Exterior Gateway Protocols	
	Distance Vector Routing Protocols	Link State Routing Protocols	Path Vector	
Classful	RIP	IGRP		EGP
Classless	RIPv2	EIGRP	OSPFv2	IS-IS
	RIPng	EIGRP for IPv6	OSPFv3	IS-IS for IPv6
				BGPv4 for IPv6

# Historique d'IGRP et EIGRP

- Développé en 1985 pour palier aux limites de RIP version 1
  - Algorithme de routage à vecteur distance utilisant une métrique en saut et une limite sur la dimension d'un réseau à 15 sauts
- Utilise les métriques suivantes :
  - bande passante (par défaut)
  - le délai (par défaut)
  - la fiabilité
  - la charge
- N'est plus supporté à partir des versions IOS 12.2(13)T et 12.2(R1s4)S
- Utilise généralement des variantes de l'algorithme Bellman-Ford ou Ford-Fulkerson
- EIGRP utilise un algorithme de diffusion appelé DUAL



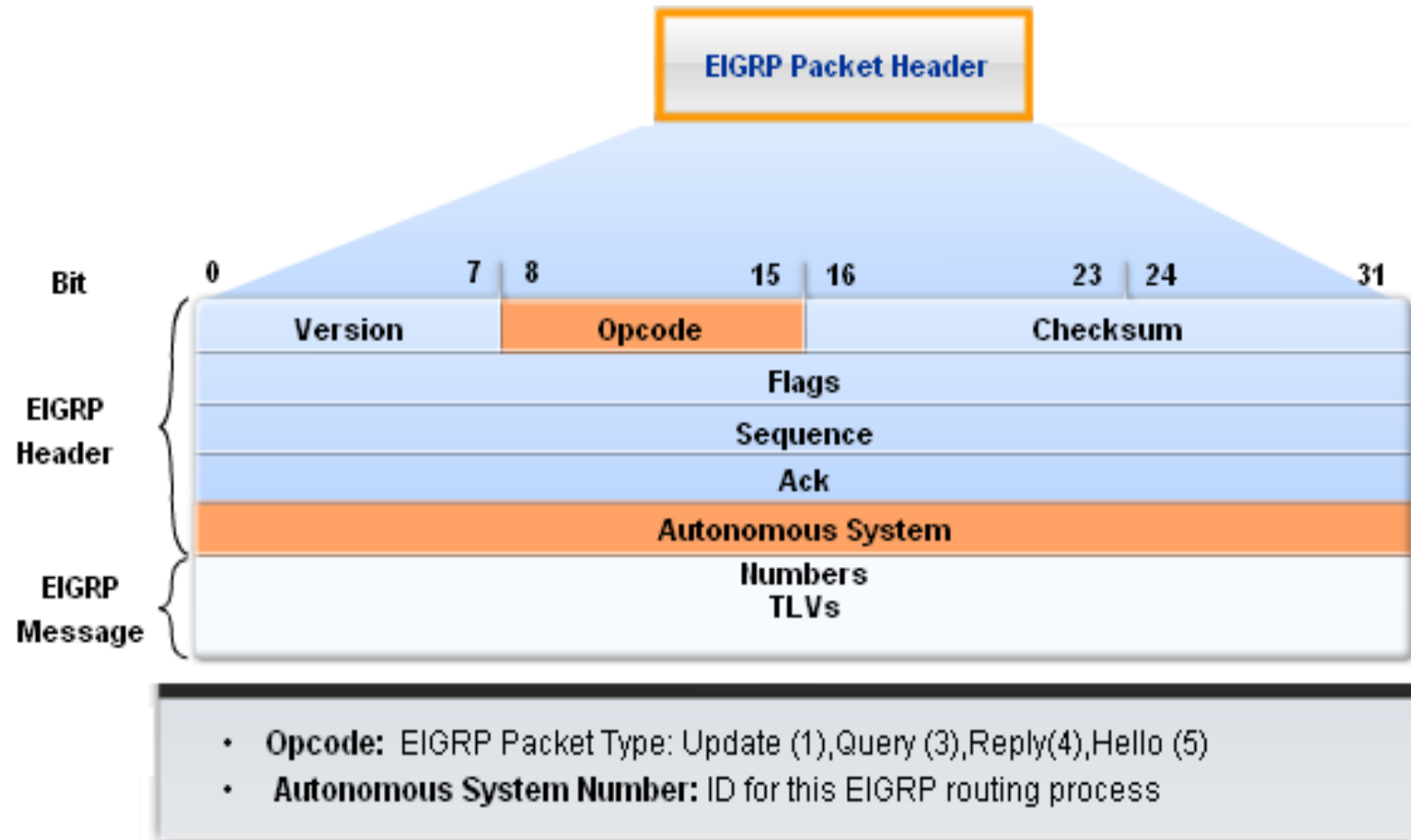
# Les messages EIGRP

- L'en-tête EIGRP contient
  - Data link Frame Header : contient les adresses MAC source et destination
  - IP Packet Header : contient les adresses IP source et destination
  - EIGRP packet header : contient les numéro d'Autonomous System (AS)
  - Type/length/Field : portion de données propre aux messages EIGRP

## Encapsulated EIGRP Message



# EIGRP packet header

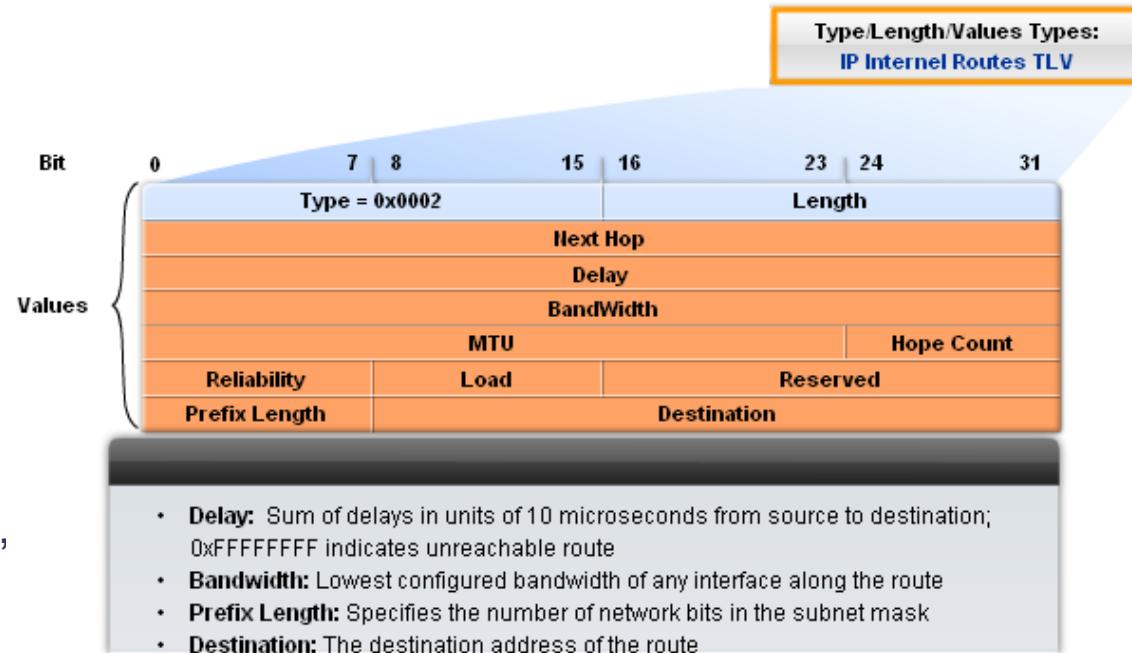


# Type/Length/Values types (TLV)



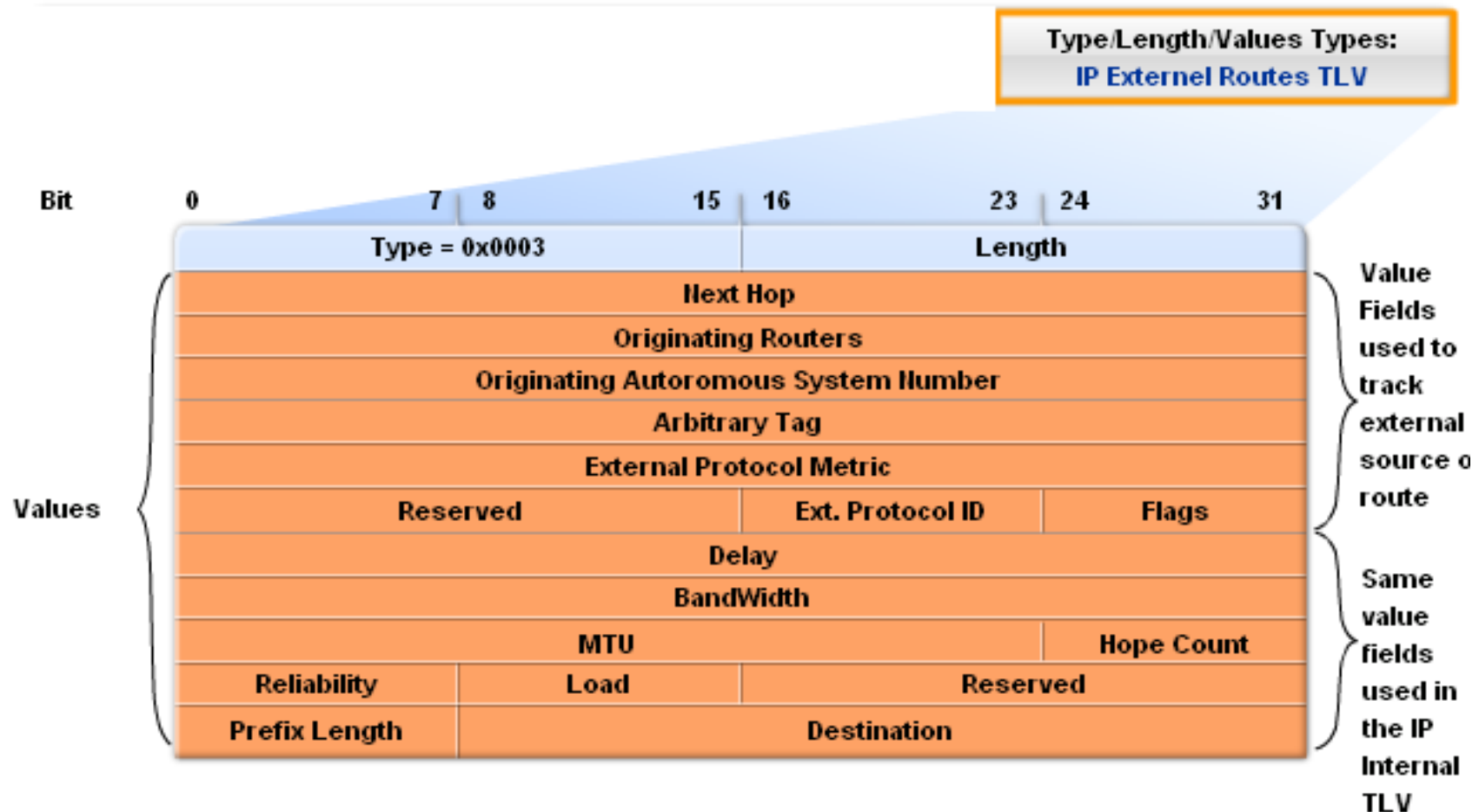
# Type/Length/Values types (TLV)

- EIGRP identifie les routes internes et externes au processus EIGRP
- TLV : IP internal contient
  - metric
  - subnet mask
  - destination
- Champ destination est de 24 bits !
  - Si besoin de plus, par exemple pour un réseau 192.168.10.192/27, 32 bits supplémentaires seront utilisés (soit 56 au total)



# TLV pour les routes externes

- TLV : IP external contient des informations utilisées quand des routes externes sont importées à l'intérieur de process EIGRP



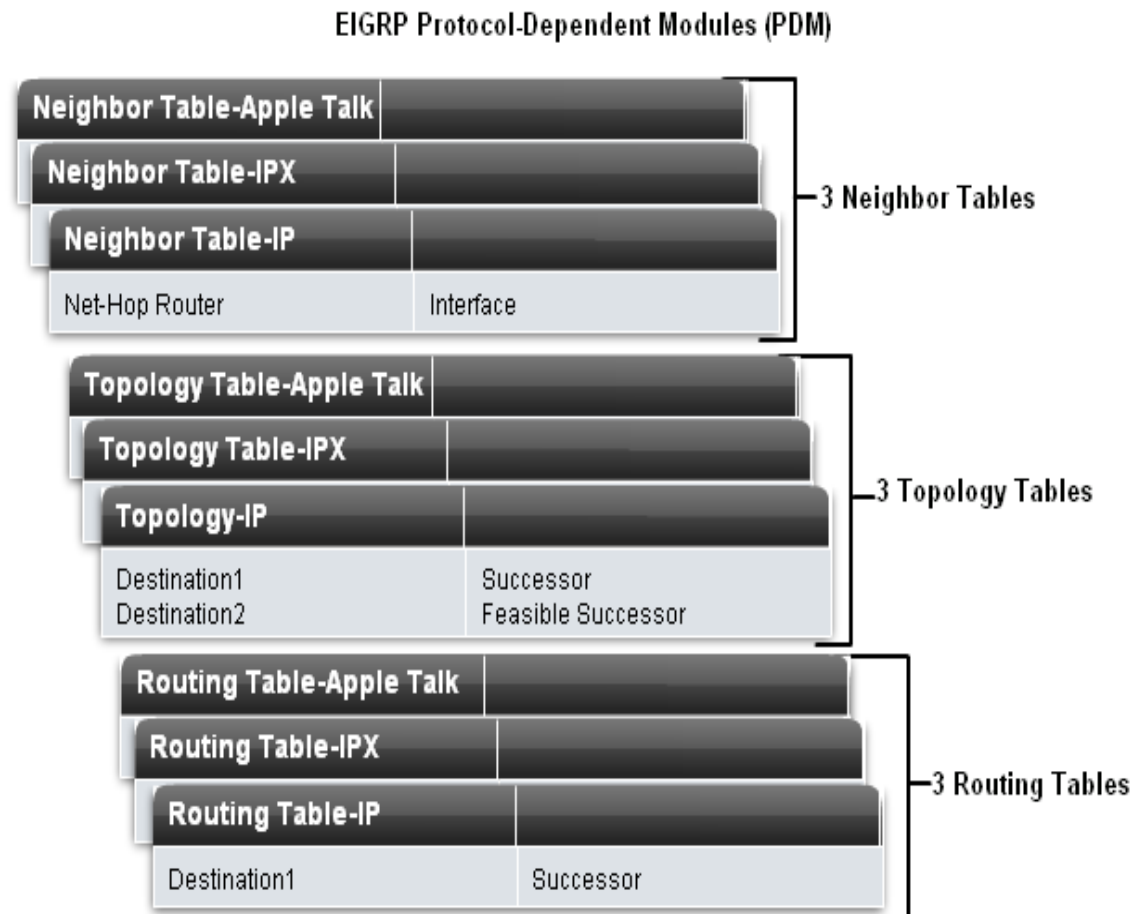
# Les tables EIGRP

- Découverte des routeurs voisins attachés à un même sous-réseau et stockage de leur identité dans une table appelé *EIGRP neighbor table*
- Echange et stockage des informations topologiques dans une table appelé *EIGRP topology table*
- Après analyse des informations topologiques, les routes de métriques les plus faibles sont stockées dans la *table de routage*



# Modules dépendants du protocole

- Comme EIGRP fonctionne à la fois avec IP, IPX et Appletalk et que chacune de ces 3 tables est dépendante du protocole réseau de couche 3 utilisé, le routeur doit maintenir constamment à jour 9 tables

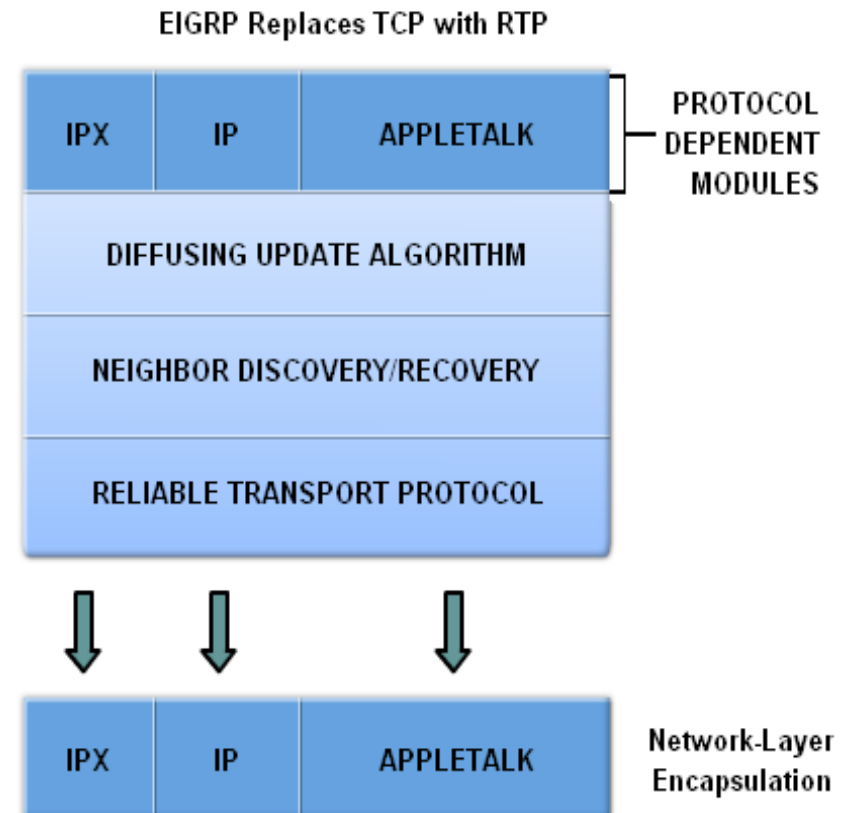


# Voisinage et information topologique

- Quand 2 routeurs se sont mutuellement découverts voisins, ils échangent complètement leur table de routage.
- Ensuite, des messages Hello sont constamment échangés afin de manifester sa présence. L'intervalle de temps séparant 2 messages Hello est par défaut de
  - 5 secondes sur un LAN ou connexion PPP
  - 60 secondes sur un WAN multi-points comme Frame Relay
- Quand une modification topologique est constatée, seules les nouveautés sont échangées,
  - par multicast à l'adresse 224.0.0.10, si plusieurs routeurs doivent être prévenus
  - par unicast dans le cas contraire
- Les mises à jour sont envoyées via le protocole RTP (Reliable Transport Protocol)

# Le protocole RTP

- Proposition de RTP
  - Utilisé par EIGRP pour les échanges de paquets EIGRP
- Caractéristiques :
  - Permet de faire à la fois de l'acheminement
    - fiable : qui nécessite des accusés réceptions
    - non fiable
  - Les paquets peuvent être envoyés
    - en unicast et
    - en multicast sur l'adresse 224.0.0.10

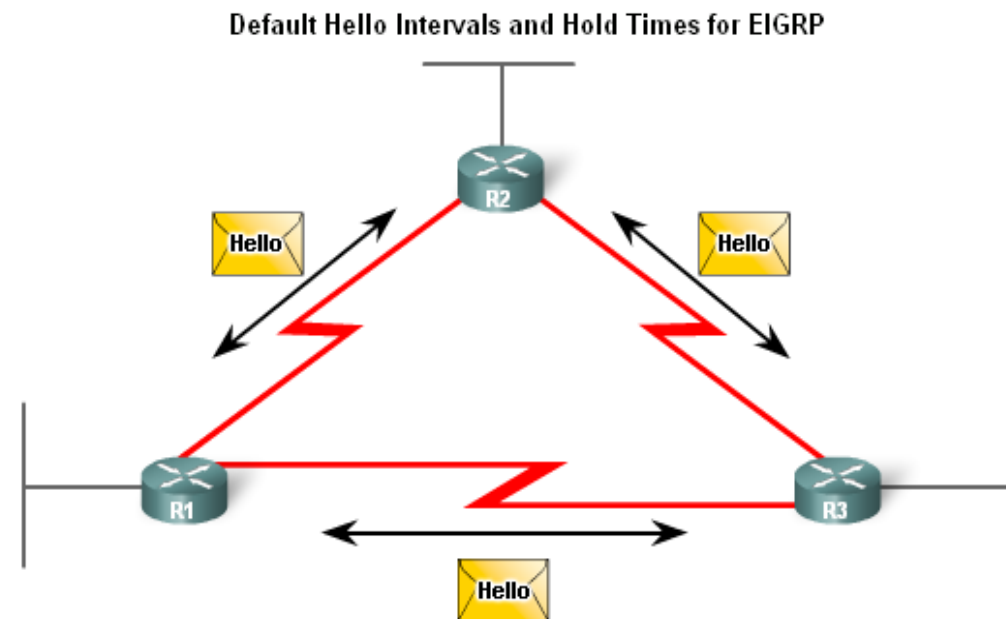


# Les types de message EIGRP

- EIGRP utilise 5 types de messages
  - Hello packets
  - Update packets
  - Acknowledgement packet
  - Query packets
  - Reply packets

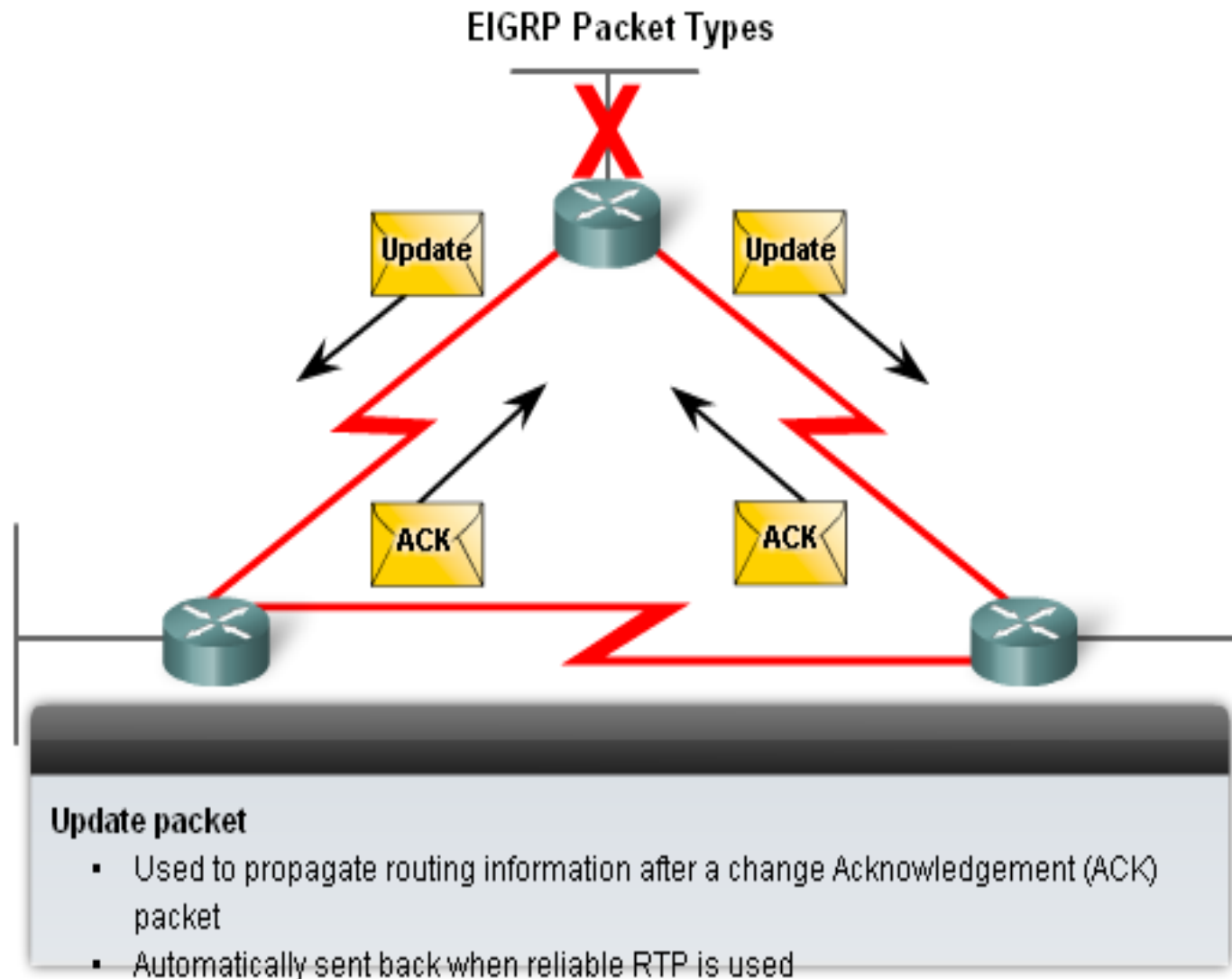
# Hello packets

- Permet de découvrir les voisins
- Envoie toutes
  - les 5 secondes sur la plupart des réseaux
  - toutes les 60 secondes sur le Non Broadcast Multi-access Networks (NBMA)
- C'est le temps maximum qu'un routeur peut attendre avant de déclarer un routeur absent
  - Holdtime
    - Par défaut : 3 fois le hello interval



Bandwidth	Example Link	Default Hello Interval	Default Hold Time
1.544 Mbps	Multipoint Frame Relay	60 seconds	180 seconds
Greater than 1.544 Mbps	T1, Ethernet	5 seconds	15 seconds

# Update packets



# EIGRP Bounded updates

- EIGRP n'envoie des mises à jour que si des changements sont constatés
- Partial update
  - N'inclut que les informations de routage ayant été modifié
- Bounded update
  - Quand une route change, seules les routeurs concernés par ce changement seront prévenus grâce à des partials updates
- EIGRP utilise des partial bounded updates pour minimiser l'utilisation de la bande passante

## EIGRP Updates

### EIGRP Updates are partial and bounded:

**Partial** because the update only includes information about route changes.

**Bounded** because only those routers affected by the change will receive the update.

# Distance administrative

- EIGRP définit 3 distances administratives différentes

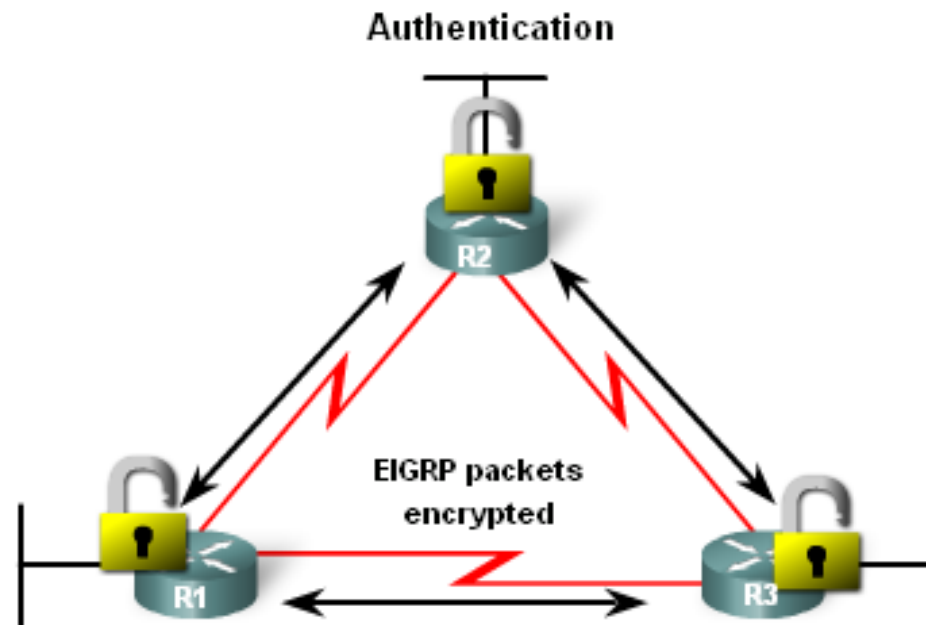
**Default Administrative Distances**

Route Source	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
Internal BGP	200



# Authentication

- EIGRP peut
  - chiffrer les informations de routage
  - authentifier les informations de routage



# Le protocole de routage EIGRP

---

Le calcul de la métrique

# La métrique

- EIGRP utilise comme métrique une association des paramètres suivants :
  - la bande passante, le délai, la fiabilité et la charge
- La formule utilisée est la suivante :

## EIGRP Composite Metric

Default Composite Formula:

$$\text{metric} = [K1 * \text{bandwidth} + K3 * \text{delay}] * 256$$

Complete Composite Formula:

$$\text{metric} = [K1 * \text{bandwidth} + (K2 * \text{bandwidth}) / (256 - \text{load}) + K3 * \text{delay}] * [K5 / (\text{reliability} + K4)] * 256$$

(Not used if "K" values are 0)

### Default values:

K1 (bandwidth) = 1

K2 (load) = 0

K3 (delay) = 1

K4 (reliability) = 0

K5 (reliability) = 0

"K" values can be changed with the **metric weights** command.

```
Router (config-router) #metric weights tos k1 k2 k3 k4 k5
```

# La métrique

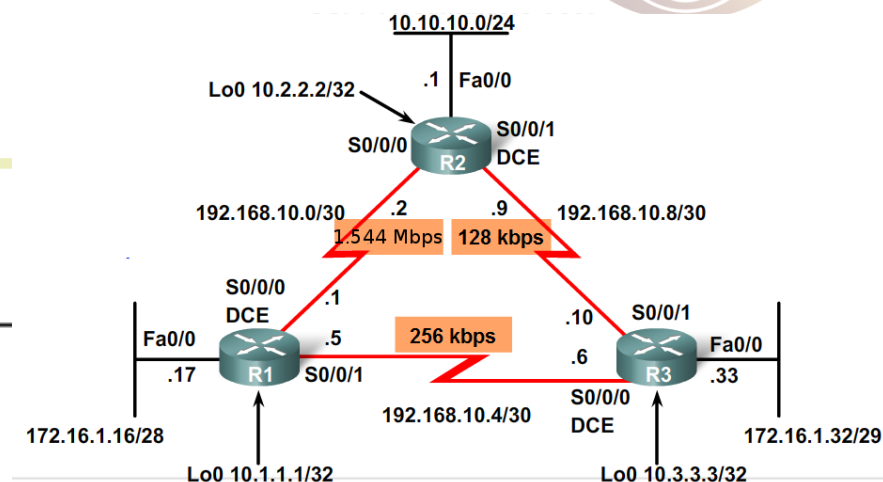
- Visualiser les paramètres K

```
R1#show ip protocols
Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  Automatic network summarization is in effect
  Automatic address summarization:
    192.168.10.0/24 for FastEthernet0/0, Serial0/0/0
      Summarizing with metric 2169856
    172.16.0.0/16 for Serial0/0/1
      Summarizing with metric 28160
  Maximum path: 4
  Routing for Networks:
    172.16.0.0
    192.168.10.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    (this router)           90           00:03:29
    192.168.10.6           90           00:02:09
    Gateway         Distance      Last Update
    172.16.3.2           90           00:02:12
  Distance: internal 90 external 170
```



# La métrique

```
R1#show interface serial 0/0/0
Serial0/0/0 is up, line protocol is up
Hardware is GT96K Serial
Description: Link to R2
Internet address is 172.16.3.1/30
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:00, output 00:00:01, output hang never
Last clearing of "show interface" counters 3d22h
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
 112522 packets input, 7303722 bytes, 0 no buffer
 Received 40016 broadcasts, 0 runts, 0 giants, 0 throttles
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
112601 packets output, 7280131 bytes, 0 underruns
 0 output errors, 0 collisions, 2 interface resets
 0 output buffer failures, 0 output buffers swapped out
12 carrier transitions
DCD=up DSR=up DTR=up RTS=up CTS=up
```



**usec = microsecond or 1 millionth of a second**

# Le délai

- Le délai est défini comme la mesure du temps de transmission d'un paquet à travers une route
  - c'est une valeur statique suivant le type de lien

**Delay Values in Microseconds**

Media	Delay
100M ATM	100 $\mu$ S
Fast Ethernet	100 $\mu$ S
FDDI	100 $\mu$ S
1HSSI	20,000 $\mu$ S
16M Token Ring	630 $\mu$ S
Ethernet	1,000 $\mu$ S
T1 (Serial Default)	20,000 $\mu$ S
512K	20,000 $\mu$ S
DSO	20,000 $\mu$ S
56K	20,000 $\mu$ S

# Les autres paramètres

- La fiabilité
  - mesuré dynamiquement
  - exprimé par une fraction de 255
  - Plus la fraction est élevée, meilleur est la fiabilité
- la charge
  - ce nombre reflète le trafic du lien
  - mesuré dynamiquement et exprimé par une fraction
  - plus cette fraction est petite, plus la charge du lien est faible et meilleur sera la métrique

## Reliability and Load Values

MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,

reliability 255/255, txload 1/255, rxload 1/255

↑  
Reliability Value

↑  
Load Value

# Modifier la bande passante

- Modifier le paramètre bande passante
  - via la commande *bandwidth* comme pour OSPF
- Vérification du paramètre
  - *Router#show interface*
- **Attention** : ce paramètre ne change pas la bande passante physique du lien correspondant



# Résultat de la métrique

## Calculating the EIGRP Default Metric

**Default metric = [K1\*bandwidth + K3\*delay] \* 256**

Since K1 and K3 both equal 1, the formula simplifies to: **bandwidth + delay**

bandwidth = speed of slowest link in route to the destination

delay = sum of the delays of each link in route to the destination

Slowest bandwidth:  $(10,000,000/\text{bandwidth kbps}) * 256$

Plus the sum of the delays:  $+ (\text{sum of delay}/10) * 256$

= EIGRP metric

```
R2#show ip route  
<output omitted>
```

```
D 192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:02:14, Serial0/0/1
```

# Le calcul

- EIGRP utilise la bande passante (BW) la plus faible dans son calcul de la métrique
  - $BW \text{ calculée} = BW \text{ de référence} / \text{la plus petite BW de la route (en kbps)}$
- Le délai EIGRP utilisé est la somme de toutes les interfaces de sortie
  - Le délai calculé = la somme de tous les délais des interfaces de sortie
- La métrique EIGRP = BW calculé + délai calculé

# Exemple

## Finding the Slowest Bandwidth

```
R2#show inter ser 0/0/1
Serial0/0/1 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 192.168.10.9/30
  MTU 1500 bytes, BW 1024 Kbit, DLY 20000 usec,
<remaining output omitted>
```

```
R3#show inter fa 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0002.b9ee.5ee0 (bia 0002.b9ee.5ee0)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
<remaining output omitted>
```

$$\text{bandwidth} = (10,000,000/1024) = 9765 * 256 = 2499840$$

# Exemple

## Summing the Delays

```
R2#show inter ser 0/0/1
Serial0/0/1 is up, line protocol is up
  Hardware is PowerQUICC Serial
  Internet address is 192.168.10.9/30
  MTU 1500 bytes, BW 1024 Kbit, DLY 20000 usec,
<remaining output omitted>

<remaining output omitted>
```

```
R3#show inter fa 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0002.b9ee.5ee0 (bia 0002.b9ee.5ee0)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
<remaining output omitted>
```

$$\text{delay} = [(20000/10) + (100/10)] * 256 = 514560$$

# Exemple

**EIGRP Metric = bandwidth + delay = 2499840 + 514560 = 3014400**

```
R2#show ip route
<code output omitted>

Gateway of last resort is not set

  192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
D    192.168.10.0/24 is a summary, 00:00:15, Null0
D    192.168.10.4/30 [90/21024000] via 192.168.10.10, 00:00:15, Serial0/0/1
C    192.168.10.8/30 is directly connected, Serial0/0/1
  172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
D    172.16.0.0/16 is a summary, 00:00:15, Null0
D    172.16.1.0/24 [90/40514560] via 172.16.3.1, 00:00:15, Serial0/0/0
C    172.16.2.0/24 is directly connected, FastEthernet0/0
C    172.16.3.0/30 is directly connected, Serial0/0/0
  10.0.0.0/30 is subnetted, 1 subnets
C    10.1.1.0 is directly connected, Loopback1
D    192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:00:15, Serial0/0/1
```

# Le protocole de routage EIGRP

---

L'algorithme DUAL

# L'algorithme DUAL

- L'élimination des boucles se fait, grâce à l'algorithme DUAL
  - **D**iffusing **U**ppdate **A**lgorithm (DUAL)

## DUAL Concepts

### DUAL provides:

- Loop-free paths
- Loop-free backup paths which can be used immediately
- Fast convergence
- Minimum bandwidth usage with bounded updates

# Suppression des boucles

- Les boucles sont supprimées grâce aux informations topologiques conservées en mémoire du routeur
- Quand plusieurs routes sont découvertes vers un même sous-réseau, celle de meilleure métrique est mise dans la table de routage et les autres sont conservées parmi les informations topologiques
- La deuxième meilleure route est conservée et est appelée « feasible successor »
  - En cas de défaillance de la meilleure route, la « feasible successor » sera alors mise dans la table de routage



# Les concepts

- L'algorithme DUAL utilise les concepts de Successor et de Feasible distance
  - Successor : identifie la meilleure route vers une destination
  - Feasible distance : la métrique la plus faible pour la route vers le réseau destination

**Feasible Distance and Successor**

```
R2#show ip route
<code output omitted>

Gateway of last resort is not set

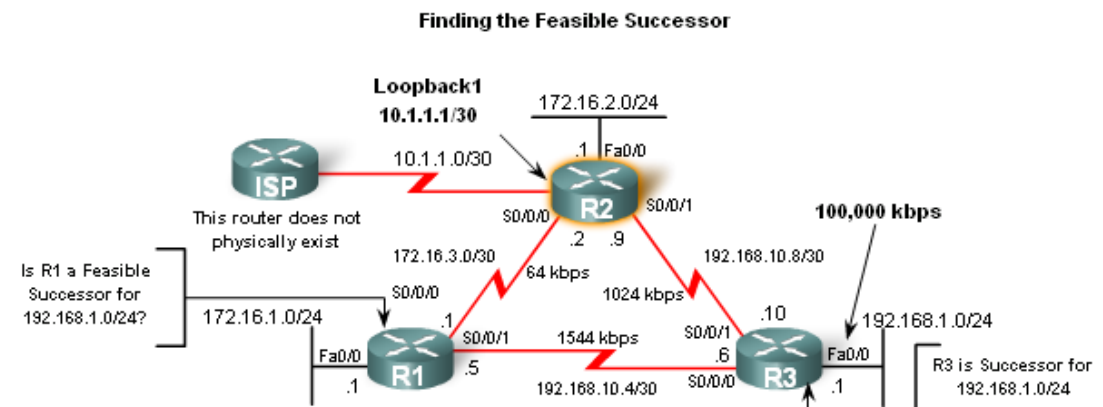
  192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
D   192.168.10.0/24 is a summary, 00:00:15, Null0
D   192.168.10.4/30 [90/21024000] via 192.168.10.10, 00:00:15,
Serial0/0/1
C   192.168.10.8/30 is directly connected, Serial0/0/1
  172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
D   172.16.0.0/16 is a summary, 00:00:15, Null0
D   172.16.1.0/24 [90/40514560] via 172.16.3.1, 00:00:15, Serial0/0/0
C   172.16.2.0/24 is directly connected, FastEthernet0/0
C   172.16.3.0/30 is directly connected, Serial0/0/0
  10.0.0.0/30 is subnetted, 1 subnets
C   10.1.1.0 is directly connected, Loopback1
D   192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:00:15, Serial0/0/1
```

feasible distance                      successor

R3 at 192.168.10.10 is the successor for network 192.168.1.0/24. This route has a feasible distance of 3014400.

# Les feasible successors ?

- C'est une route « secondaire », sans boucle, vers la même destination que la successor route
- Pour être feasible successor, il faut satisfaire la feasible (ou feasibility) condition
- Cette condition se rapporte à une distance appelée la **reported distance** ou **advertised distance**



```

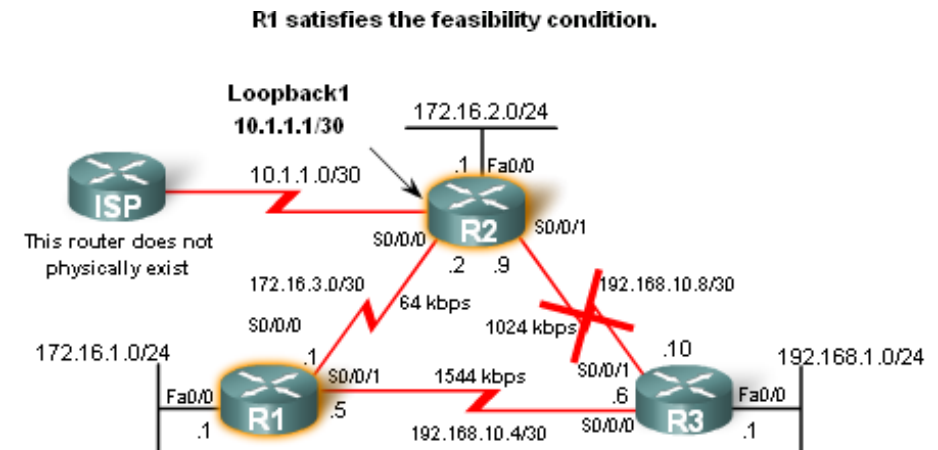
R2#show ip route
<code output omitted>

Gateway of last resort is not set

  192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
D   192.168.10.0/24 is a summary, 00:00:15, Null0
D   192.168.10.4/30 [90/21024000] via 192.168.10.10, 00:00:15, Serial0/0/1
C   192.168.10.8/30 is directly connected, Serial0/0/1
  172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
D   172.16.0.0/16 is a summary, 00:00:15, Null0
D   172.16.1.0/24 [90/40514560] via 172.16.3.1, 00:00:15, Serial0/0/0
C   172.16.2.0/24 is directly connected, FastEthernet0/0
C   172.16.3.0/30 is directly connected, Serial0/0/0
  10.0.0.0/30 is subnetted, 1 subnets
C   10.1.1.0 is directly connected, Loopback1
D   192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:00:15, Serial0/0/1
  
```

# Reported ou advertised distance (RD) ?

- Egalement appelée Advertised Distance (AD)
- C'est la feasible distance envoyé par un voisin d'un routeur, vers une destination
- Dans l'exemple, la RD vers 192.168.1.0/24 envoyé par R1 à R2 est 2172416



```
R2#show ip route
<output omitted for brevity>

D   192.168.1.0/24 [90/3014400] via 192.168.10.10, 00:00:15, Serial10/0/1
-----

R1#show ip route
<output omitted for brevity>

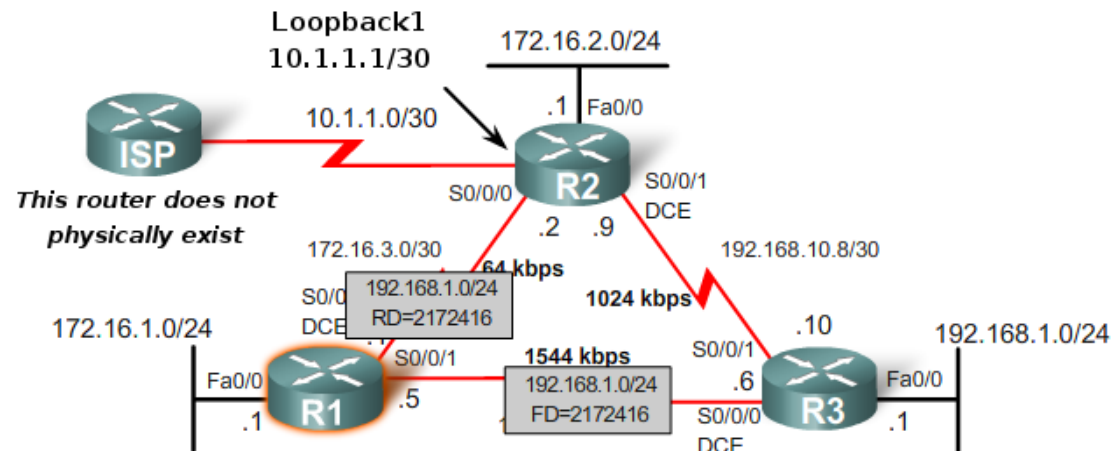
D   192.168.1.0/24 [90/2172416] via 192.168.10.6, 01:12:26, Serial10/0/1
```

# Feasibility condition ?

- Condition satisfaite par un voisin dont la RD est inférieure à la FD du routeur vers la même destination

- R1 va donc devenir un feasible successor pour aller de R2 à 192.168.1.0/24

Does R1 satisfy the feasibility condition?



```
R1#show ip route
<output omitted for brevity>

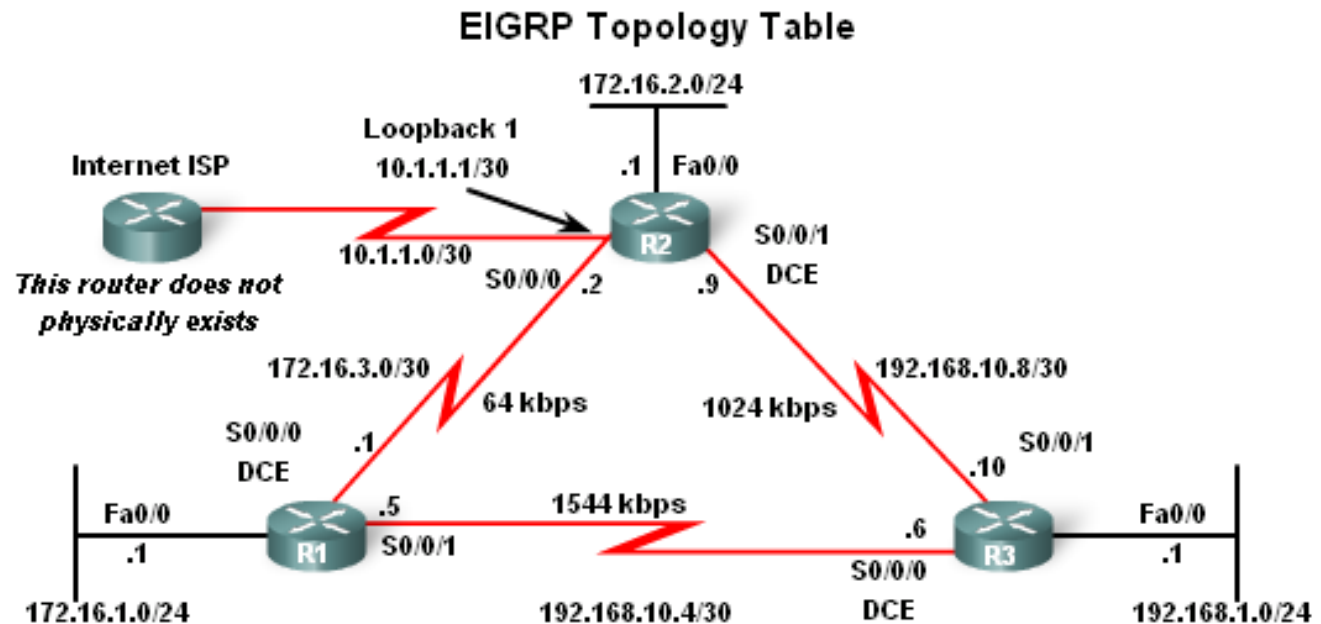
D    192.168.1.0/24 [90/2172416] via 192.168.10.6, 01:12:26, Serial10/0/1
```

R1 reports to R2 that its feasible distance to 192.168.1.0/24 is  
2172416

# Topology table

- La commande `show ip eigrp topology` permet de visualiser :

- les successor routes
- les feasible successor routes



```
R2#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.1.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R -
Reply,
       r - reply Status, s - sia Status

<output omitted>
P 192.168.1.0/24, 1 successors, FD is 3014400
   via 192.168.10.10 (3014400/28160), Serial10/0/1
   via 172.16.3.1 (41026560/2172416), Serial0/0/0
P 192.168.10.8/30, 1 successors, FD is 3011840
   via Connected, Serial0/1
```

**Routes for 192.168.1.0/24**

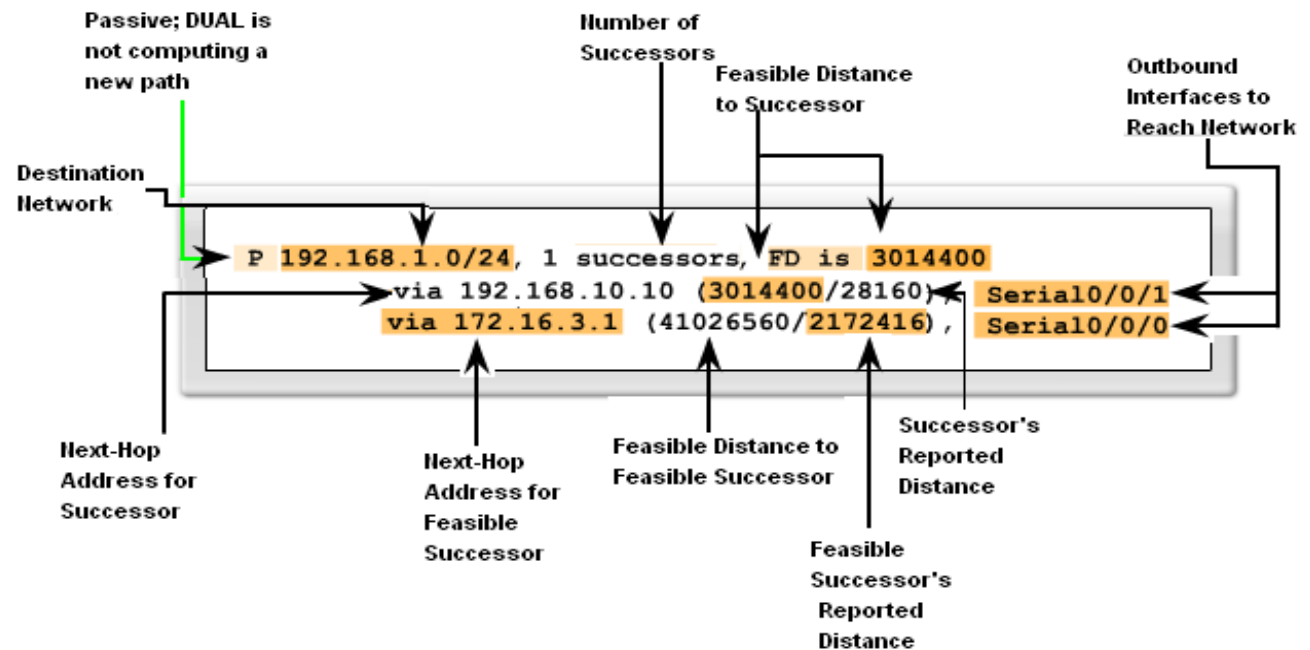
# Exemple de table de topologie

```
R2#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(10.1.1.1)

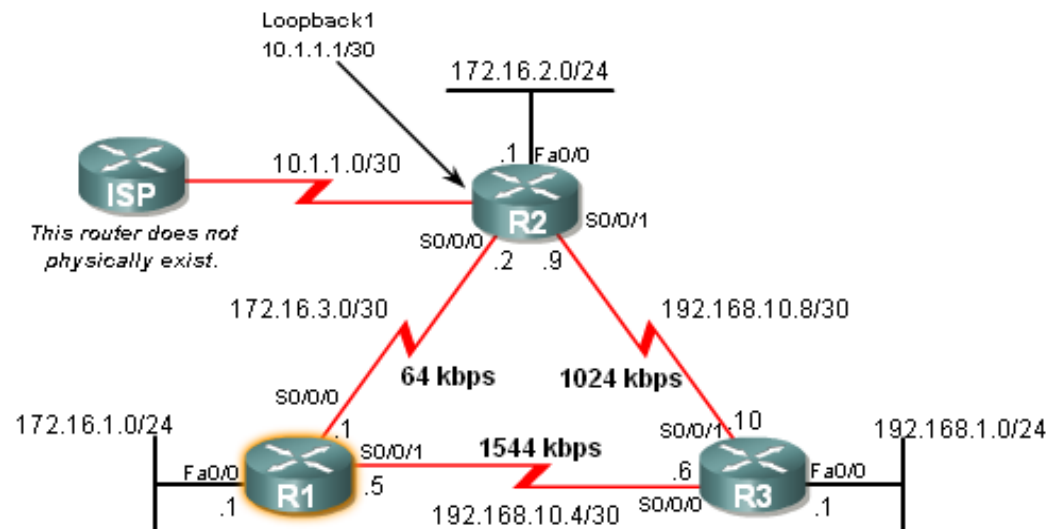
Codes: P - Passive, A - Active, U - Update, Q - Query, R -
Reply,
      r - reply Status, s - sia Status

<output omitted>
P 192.168.1.0/24, 1 successors, FD is 3014400
  via 192.168.10.10 (3014400/28160), Serial0/0/1
  via 172.16.3.1 (41026560/2172416), Serial0/0/0
```

Table Entry for 192.168.1.0/24



# No feasible successor ?



## No Feasible Successor

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(192.168.10.5)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 192.168.10.0/24, 1 successors, FD is 2169856
   via Summary (2169856/0), Null0
P 192.168.10.4/30, 1 successors, FD is 2169856
   via Connected, Serial0/0/1
P 192.168.1.0/24, 1 successors, FD is 2172416
   via 192.168.10.6 (2172416/28160), Serial0/0/1
P 192.168.10.8/30, 1 successors, FD is 3523840
   via 192.168.10.6 (3523840/3011840), Serial0/0/1
<output omitted>
```

```
R1#show ip eigrp topology all-links
IP-EIGRP Topology Table for AS(1)/ID(192.168.10.5)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 192.168.10.0/24, 1 successors, FD is 2169856, serno 3
   via Summary (2169856/0), Null0
   via 172.16.3.2 (41024000/3011840), Serial0/0/0
P 192.168.10.4/30, 1 successors, FD is 2169856, serno 1
   via Connected, Serial0/0/1
P 192.168.1.0/24, 1 successors, FD is 2172416, serno 5
   via 192.168.10.6 (2172416/28160), Serial0/0/1
   via 172.16.3.2 (41026560/3014400), Serial0/0/0
P 192.168.10.8/30, 1 successors, FD is 3523840, serno 11
   via 192.168.10.6 (3523840/3011840), Serial0/0/1
<output omitted>
```

RD from R2 is higher than FD to R1.

# Le DUAL actif

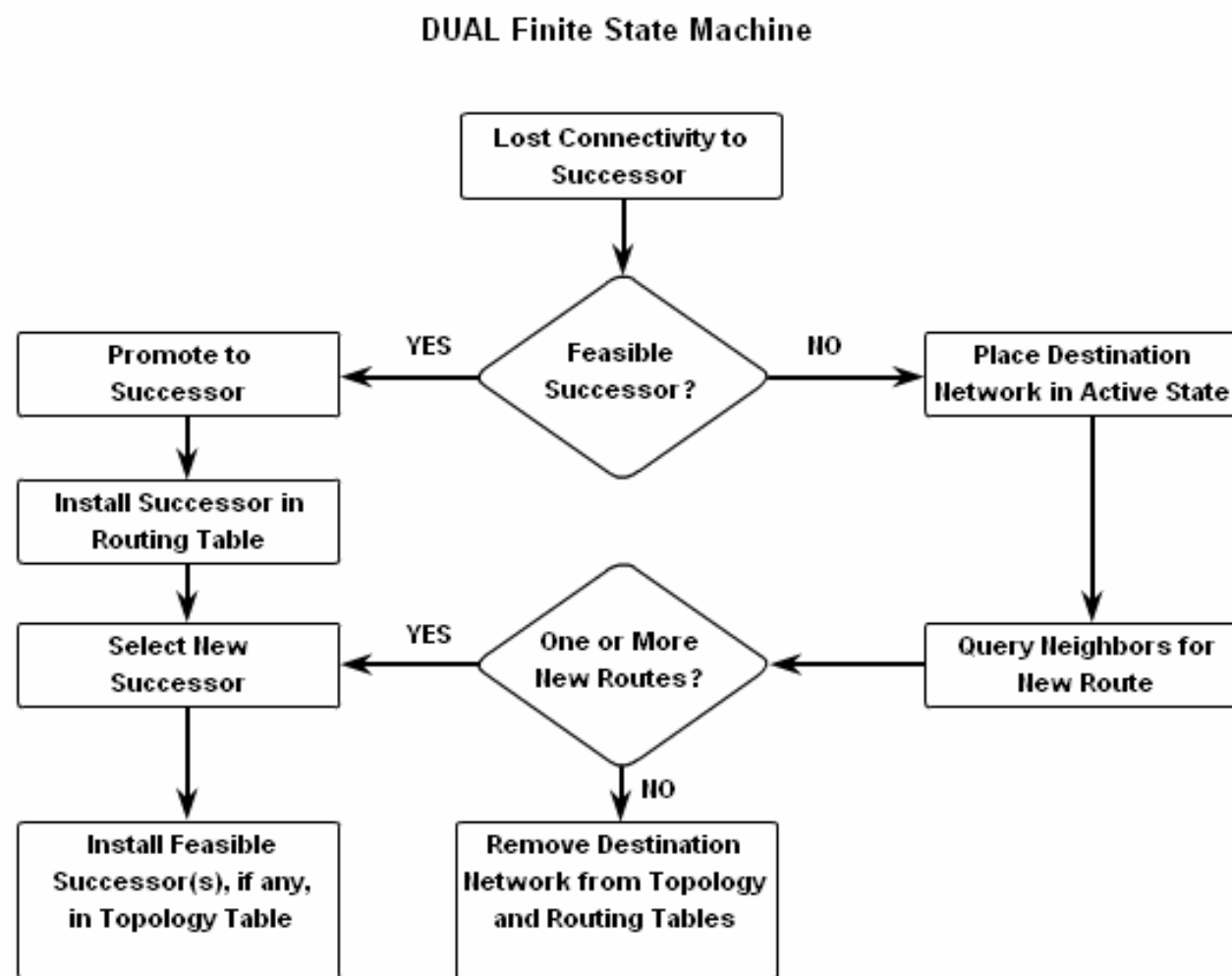
- Un feasible successor peut ne pas être choisi, simplement car la feasibility condition n'est pas remplie
  - Cela signifie que la RD vers une destination, rapporté par un voisin, est supérieure ou égale à l'actuelle FD
- Une recherche de route secondaire va donc se faire grâce aux messages Query
- Dans ce cas, la route est dite active



# La Finite State Machine (FSM)

- C'est une machine abstraite qui définit les états possibles d'un routeur EIGRP
- FSM est utilisée pour définir la façon comment les « device » fonctionnent en fonction des événements reçus

# La FSM de l'algorithme DUAL



# Etats de la FSM

- Pour visualiser les états de la FSM relative à EIGRP, il faut utiliser la commande *debug eigrp fsm*

```
R2#debug eigrp fsm
EIGRP FSM Events/Actions debugging is on
R2#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R2(config)#int s0/0/1
R2(config-if)#shutdown
<some debug output omitted>

DUAL: Find FS for dest 192.168.1.0/24. FD is 3014400, RD is 3014400
DUAL: 192.168.10.10 metric 4294967295/4294967295
DUAL: 172.16.3.1 metric 41026560/2172416 found Dmin is 41026560
DUAL: Removing dest 192.168.1.0/24, nexthop 192.168.10.10
DUAL: RT installed 192.168.1.0/24 via 172.16.3.1

R2(config-if)#end
R2#undebug all
All possible debugging has been turned off

R2#show ip route
<some output omitted>

D    192.168.1.0/24 [ 90/41026560] via 172.16.3.1, 00:08:58, Serial0/0
```

# Le protocole de routage EIGRP

---

- L'auto-summarization

# La route Null0

- Par défaut, EIGRP, utilise l'interface Null0 pour supprimer un paquet qui vérifie les 2 règles suivantes :
  - Il correspond bien à une route d'un réseau parent
  - Et à aucune route du même sous-réseau

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

 192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
D   192.168.10.0/24 is a summary, 00:45:09, Null0
C   192.168.10.4/30 is directly connected, Serial0/0/1
D   192.168.10.8/30 [90/3523840] via 192.168.10.6, 00:44:56, Serial0/0/1
 172.16.0.0/16 is variably subnetted, 4 subnets, 3 masks
D   172.16.0.0/16 is a summary, 00:46:10, Null0
C   172.16.1.0/24 is directly connected, FastEthernet0/0
D   172.16.2.0/24 [90/40514560] via 172.16.3.2, 00:45:09, Serial0/0/0
C   172.16.3.0/30 is directly connected, Serial0/0/0
D   192.168.1.0/24 [90/2172416] via 192.168.10.6, 00:44:55, Serial0/0/1
```

**EIGRP installs a Null0 summary route for each parent route.  
Packets matching the Null0 summary route are discarded.**

# Null0 Summary Route

- Avec un routage classless et sans la Null0 Summary Route, dans l'exemple précédent, les paquets à destination de sous-réseaux de 172.16.0.0 autres que les sous-réseaux 172.16.1.0/24, 172.16.2.0/24 ou 172.16.3.0/24 seront supprimés
- EIGRP inclut automatiquement un **Null0 summary route** pour une route si les 2 conditions suivantes existent :
  - Il existe au moins un sous-réseau appris via EIGRP
  - La fonction d'auto-summarization est activée
- Pour désactiver l'auto-summarization, il faut utiliser la commande `no auto-summary`

# Le protocole de routage EIGRP

---

## Configuration

# Configuration d'EIGRP

- La configuration se fait de façon similaire à celle d'OSPF
- Par contre, besoin d'un identifiant appelé Autonomous System (AS)
- Chaque système est identifié par un numéro d'AS, attribué par l'IANA
- Ce paramètre n'est actuellement utilisé que par l'algorithme BGP
  - Les autres algorithmes utilisent, à la place, un identifiant de process ID



# La commande network

- La configuration de EIGRP se fait de façon similaire à celle de OSPF

```
Router(config)# router eigrp AS-number
```

```
Router(config-router)# network network-address
```

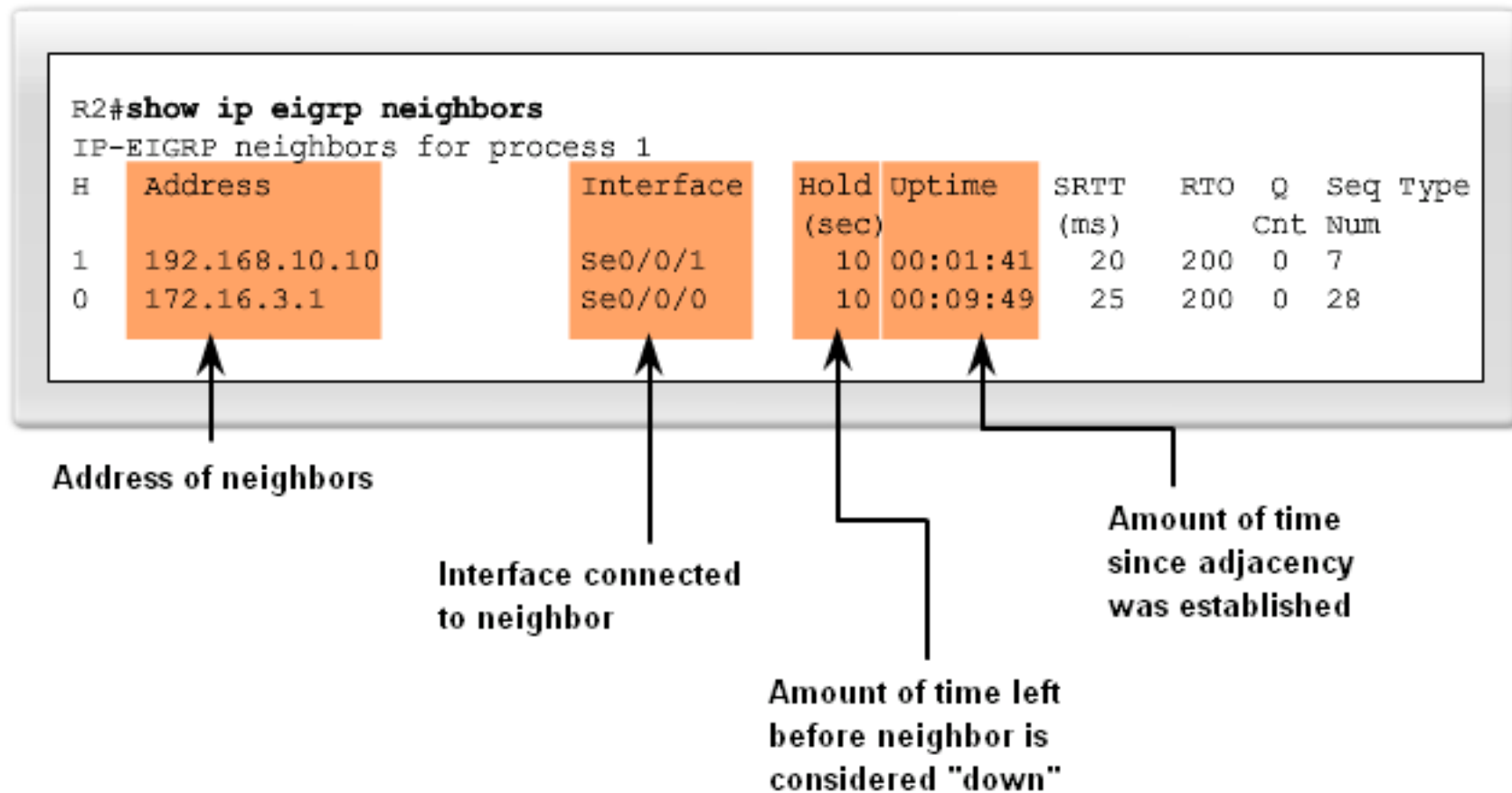
ou bien

```
Router(config)# router eigrp AS-number
```

```
Router(config-router)# network network-address wildcard-mask
```

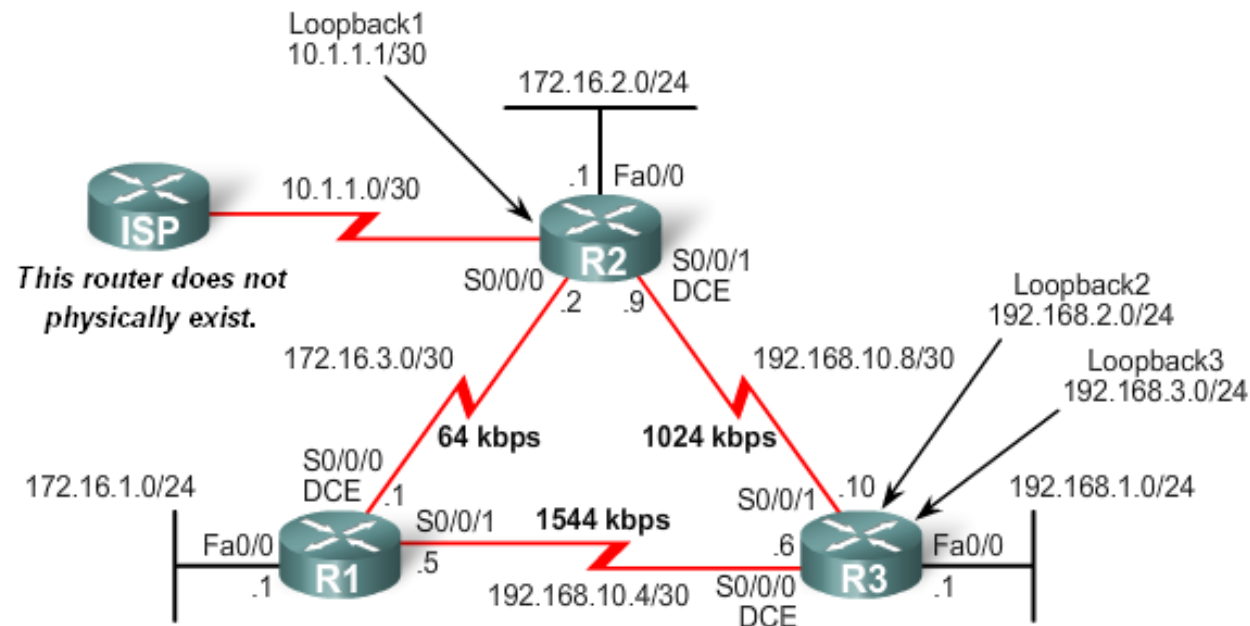
- Par défaut, une « automatic summarization » est effectué par EIGRP
  - no auto-summary est nécessaire pour désactiver cette fonction

# Visualiser les voisins



# Manual summarization

- Il est possible de faire les agrégation des routes de façon manuelle
- Ici, les routeurs R2 et R3 vont apprendre 3 routes vers 192.168.1.0/24, 192.168.2.0/24 et 192.168.3.0/24
- Possibilité d'imposer une annonce vers R2 et R3 uniquement de 192.168.0.0/22



# Manual summarization

- Dans l'exemple précédent, pour annoncer sur l'interface S0/0/0 et S0/0/1

```
R3(config)#int serial 0/0/0
```

```
R3(config-if)#ip summary-address eigrp 1 192.168.0.0 255.255.252.0
```

```
R3(config)#int serial 0/0/1
```

```
R3(config-if)#ip summary-address eigrp 1 192.168.0.0 255.255.252.0
```

- On obtient alors le résultat de la diapositive suivante

# Résultat d'une Manual summarization

```
R1#show ip route
<output omitted>

Gateway of last resort is not set

192.168.10.0/30 is subnetted, 2 subnets
C    192.168.10.4 is directly connected, Serial0/0/1
D    192.168.10.8 [90/3523840] via 192.168.10.6, 00:01:34, Serial0/0/1
172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C    172.16.1.0/24 is directly connected, FastEthernet0/0
D    172.16.2.0/24 [90/3526400] via 192.168.10.6, 00:01:12, Serial0/0/1
C    172.16.3.0/30 is directly connected, Serial0/0/0
D    192.168.0.0/22 [90/2172416] via 192.168.10.6, 00:01:11, Serial0/0/1
```

```
R2#show ip route
<output omitted>

Gateway of last resort is not set

192.168.10.0/30 is subnetted, 2 subnets
D    192.168.10.4 [90/3523840] via 192.168.10.10, 00:00:23, Serial0/0/1
C    192.168.10.8 is directly connected, Serial0/0/1
172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
D    172.16.1.0/24 [90/3526400] via 192.168.10.10, 00:00:23, Serial0/0/1
C    172.16.2.0/24 is directly connected, FastEthernet0/0
C    172.16.3.0/30 is directly connected, Serial0/0/0
10.0.0.0/30 is subnetted, 1 subnets
C    10.1.1.0 is directly connected, Loopback1
D    192.168.0.0/22 [90/3014400] via 192.168.10.10, 00:00:23, Serial0/0/1
```

# Redistribution des routes statiques

---

- La route par défaut 0.0.0.0 est indépendante du protocole de routage
- Comme c'est une route statique, il faut faire « redistribuer » cette route par le processus EIGRP grâce à la commande à ajouter dans la configuration d'EIGRP

`redistribute static`