



## Fiche TD6 : exercices 3,5,6,7,8

---

Olivier GLÜCK

Université LYON 1 / Département Informatique

Olivier.Gluck@univ-lyon1.fr

<http://perso.univ-lyon1.fr/olivier.gluck>

### 3. Proxy ARP

- Un site d'une entreprise interconnecte ces 4 personnes :

<u>station</u>	<u>@IP</u>	<u>@MAC</u>
Pierre	10.99.43.27	MAC_1
Jacques	10.163.12.254	MAC_2
Alfred	10.189.12.27	MAC_3
Martine	10.126.43.254	MAC_4

- Quel est le NET\_ID de ce plan d'adressage ?
- Quel est le nombre de bits nécessaires pour réaliser deux sous-réseaux : un regroupant Pierre et Martine, l'autre Jacques et Alfred ?  
Donnez le masque correspondant. Même question pour qu'aucune des stations n'appartiennent au même sous-réseau.
- On se place dans le cas des 2 sous-réseaux, Pour permettre la communication entre les 4 machines, quel équipement est nécessaire ? Affectez des adresses à cet équipement et faites un schéma représentant l'ensemble du réseau. Que faut-il configurer sur les machines et le routeur pour que tous les ping fonctionnent ? Décrire les trames envoyées lors d'un ping de P vers M et de P vers A.
- Cet équipement est maintenant configuré en proxy ARP. Indiquez le contenu du cache ARP de Pierre si toutes les stations ont déjà communiqué entre elles.
- L'entreprise envisage de raccorder son site à Internet. Dans quelle mesure cela est-il possible ?

### 3. Proxy ARP

<u>station</u>	<u>@IP</u>	<u>@MAC</u>
Pierre (P)	10. <b>99</b> .43.27	MAC_1
Jacques (J)	10. <b>163</b> .12.254	MAC_2
Alfred (A)	10. <b>189</b> .12.27	MAC_3
Martine (M)	10. <b>126</b> .43.254	MAC_4

- Quel est le NET\_ID de ce plan d'adressage ?

Le NET\_ID correspond à la partie réseau de l'adresse IP. Ici, les adresses commencent par **10** donc il s'agit d'adresses de classe A (car le bit le plus à gauche soit  $2^31=0$ ) donc le netmask est /8 donc NET\_ID correspond au 1<sup>er</sup> octet de l'adresse : **NET\_ID=10**.

Rq : 10.0.0.0/8 correspond aux adresses IP **privées** de classe A

- Quel est le nombre de bits nécessaires pour réaliser deux sous-réseaux : un regroupant Pierre et Martine, l'autre Jacques et Alfred ?  
Donnez le masque correspondant.

Pour faire deux sous-réseaux (SR), il faut 1 bit donc le masque des SR devient /8+1=/9. Le bit utilisé pour les 2 SR est le bit  $2^7=128$  dans le 2<sup>ème</sup> octet.

$$\begin{array}{c} \text{8 bits + 1 bit} \quad \quad \quad \text{7 bits + 8 bits + 8 bits pour les machines} \\ \hline \text{/9} = 255 . \boxed{2^7} 2^6 2^5 2^4 2^3 2^2 2^1 2^0 . 0 . 0 \end{array}$$

Les 2 SR sont donc **SR1=10.0.0.0/9** et **SR2=10.128.0.0/9**  
P et M sont bien dans SR1, J et A dans SR2 donc tout est ok

### 3. Proxy ARP

<u>station</u>	<u>@IP</u>	<u>@MAC</u>
Pierre (P)	10. <b>99</b> .43.27	MAC_1
Jacques (J)	10. <b>163</b> .12.254	MAC_2
Alfred (A)	10. <b>189</b> .12.27	MAC_3
Martine (M)	10. <b>126</b> .43.254	MAC_4

- Pour qu'aucune des stations n'appartiennent au même sous-réseau :  
Il faut découper en SR de taille plus petite jusqu'à séparer P, J, A, M.  
Pour faire 4 SR, il faut 2 bits donc le masque des SR devient  $/8+2=/10$  Le bit de poids faible (le plus à droite) dans la partie réseau est le bit  $2^6=64$  dans le 2<sup>ème</sup> octet :  

8 bits + 2 bits
|
6 bits + 8 bits + 8 bits pour les machines

$$/10 = 255 \cdot 2^7 \cdot 2^6 \cdot 2^5 \cdot 2^4 \cdot 2^3 \cdot 2^2 \cdot 2^1 \cdot 2^0 \cdot 0 \cdot 0$$

Les 4 SR sont les multiples de 64 dans le 2<sup>ème</sup> octet donc

SR1=10.0.0.0/10 SR2=10.64.0.0/10 SR3=10.128.0.0/10 SR4=10.192.0.0/10

P et M sont dans SR2, J et A sont dans SR3 donc il faut encore découper pour les séparer. Avec /11, 8 SR qui sont les multiples de 32 :

SR1=10.0.0.0/11 SR2=10.32.0.0/11 SR3=10.64.0.0/11 SR4=10.96.0.0/11

SR5=10.128.0.0/11 SR6=10.160.0.0/11 SR7=10.192.0.0/11 SR8=10.224.0.0/11

P et M sont dans SR4, J et A sont dans SR6 donc il faut encore découper pour les séparer. Avec /12, 16 SR qui sont les multiples de 16 : c'est bon car  $10.96 < P < 10.112 < M < 10.128$  et  $10.160 < J < 10.176 < A < 10.192$

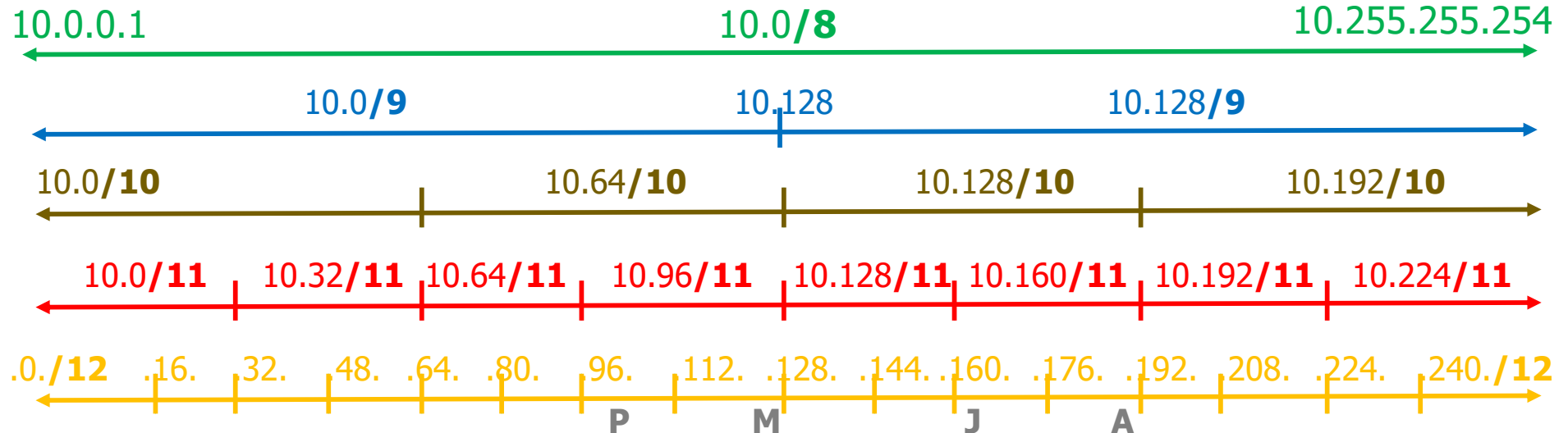
### 3. Proxy ARP

station	@IP	@MAC
Pierre (P)	10. <b>99</b> .43.27	MAC_1
Jacques (J)	10. <b>163</b> .12.254	MAC_2
Alfred (A)	10. <b>189</b> .12.27	MAC_3
Martine (M)	10. <b>126</b> .43.254	MAC_4

- Pour qu'aucune des stations n'appartiennent au même sous-réseau :

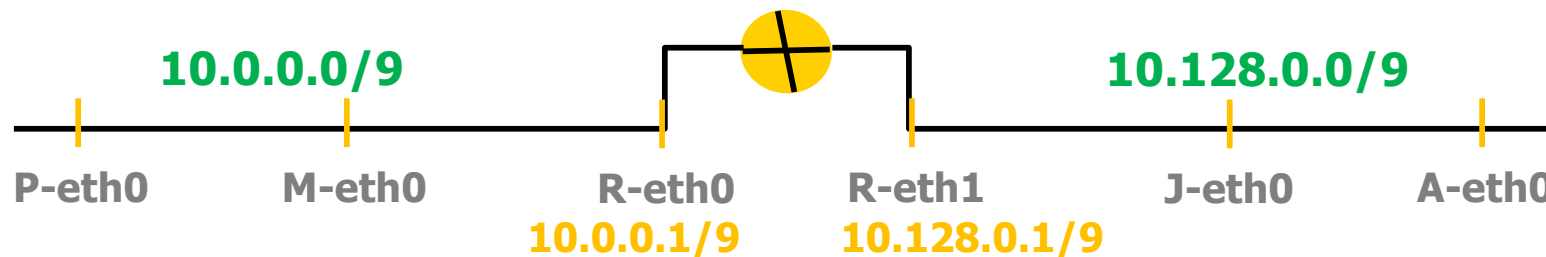
Avec /12, 16 SR qui sont les multiples de 16 : c'est bon car  
 $10.96 < P < 10.112 < M < 10.128$  et  $10.160 < J < 10.176 < A < 10.192$

Vision graphique des 16 sous-réseaux en /12 :



### 3. Proxy ARP

- On se place dans le cas des 2 sous-réseaux. Pour permettre la communication entre les 4 machines, quel équipement est nécessaire ? Affectez des adresses à cet équipement et faites un schéma représentant l'ensemble du réseau.



- Que faut-il configurer sur les machines et le routeur pour que tous les ping fonctionnent ? Décrire les trames envoyées lors d'un ping de P vers M et de P vers A.

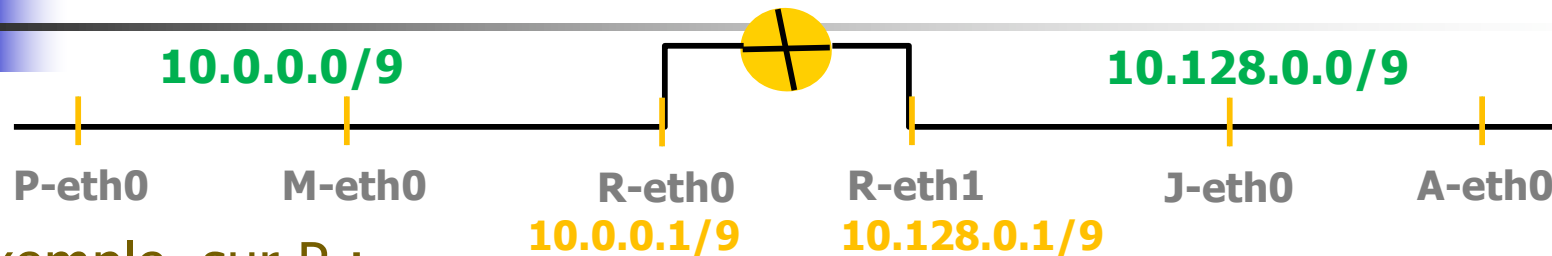
Il faut configurer la carte réseau des 4 machines et les deux du routeur.

Cela permet à chaque machine de pinguer les interfaces du même SR.

Sur chaque machine, il faut ajouter une ligne dans sa table de routage pour joindre l'autre sous-réseau. Par exemple, sur P :

```
ip a add 10.99.43.27/9 dev eth0 ; ip link set up dev eth0  
ip route add 10.128.0.0/9 via 10.0.0.1
```

### 3. Proxy ARP



Par exemple, sur P :

```
ip a add 10.99.43.27/9 dev eth0 ; ip link set up dev eth0
```

```
ip route add 10.128.0.0/9 via 10.0.0.1
```

**ip route** affiche la table de routage de P qui contient alors :

Destination/masque	Prochain saut ou Interface
10.0.0.0/9	eth0
10.128.0.0/9	10.0.0.1

Par exemple, sur A :

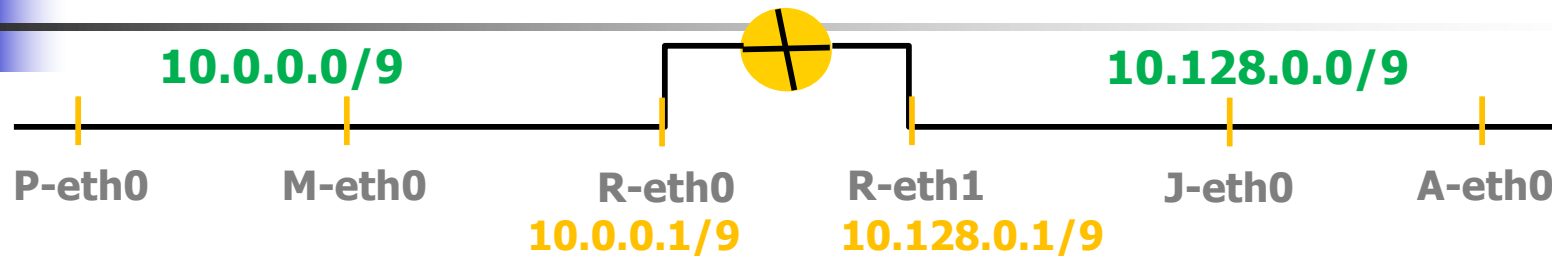
```
ip a add 10.189.12.27/9 dev eth0 ; ip link set up dev eth0
```

```
ip route add 10.0.0.0/9 via 10.128.0.1
```

**ip route** affiche la table de routage de A qui contient alors :

Destination/masque	Prochain saut ou Interface
10.128.0.0/9	eth0
10.0.0.0/9	10.128.0.1

### 3. Proxy ARP



- Décrire les trames envoyées lors d'un ping de P vers M.

Sur P : **ping 10.126.43.254**

Fabrication d'une requête ICMP encapsulée dans IP et Ethernet :

@IPdst=10.126.43.254 @MACdst=? @IPsrc=10.99.43.27 @MACsrc=P-eth0

Pour cette destination, la table de routage de P dit qu'il faut passer par P-eth0 ce qui signifie que le destinataire est dans le SR de P-eth0 donc

Fabrication d'une requête ARP encapsulée dans Ethernet pour trouver l'@MAC de 10.126.43.254 (M-eth0) : @MACdst=FF:FF:FF:FF:FF:FF @MACsrc=P-eth0

Envoi de la requête ARP par P. Fabrication et envoi de la réponse ARP par M :

@MACdst=P-eth0 @MACsrc=M-eth0 type=0806

Le ? est maintenant connu.

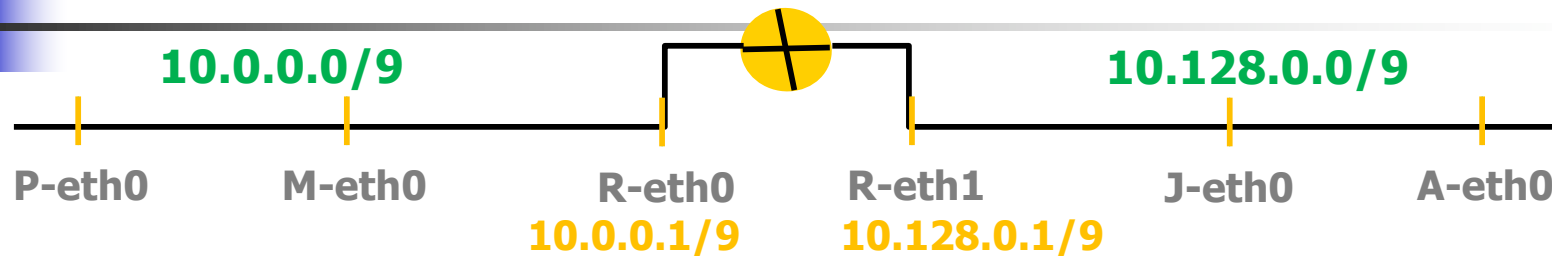
Envoi de la requête ICMP par P.

Envoi de la réponse ICMP par M. (on suppose ici que M connaît @MAC de P)

Il faut donc 4 trames pour faire un ping-pong entre P et M : 2 ARP + 2 ICMP



### 3. Proxy ARP



- Décrire les trames envoyées lors d'un ping de P vers A.

Sur P : **ping 10.189.12.27**

Fabrication d'une requête ICMP encapsulée dans IP et Ethernet :

@IPdst=10.189.12.27 @MACdst=? @IPsrc=10.99.43.27 @MACsrc=P-eth0

Pour cette destination, la table de routage de P dit qu'il faut passer par 10.0.0.1 mais l'adresse MAC de R-eth0 est inconnue donc

Fabrication d'une requête ARP encapsulée dans Ethernet pour trouver l'@MAC de 10.0.0.1 (R-eth0) : @MACdst=FF:FF:FF:FF:FF:FF @MACsrc=P-eth0

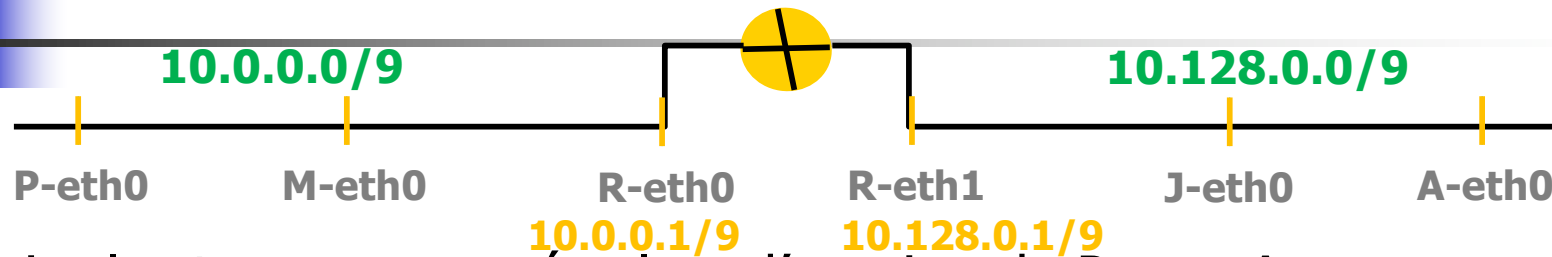
Envoi de la requête ARP par P. Fabrication et envoi de la réponse ARP par R :

@MACdst=P-eth0 @MACsrc=R-eth0 type=0806

Le ? est maintenant connu.

Envoi de la requête ICMP par P. Le routeur reçoit la trame, décapsule, récupère l'@IPdst dans l'en-tête IP= 10.189.12.27, cherche dans sa table de routage comment atteindre cette destination : il faut passer par l'interface R-eth1 car le routeur est directement connecté au réseau de la destination

### 3. Proxy ARP



- Décrire les trames envoyées lors d'un ping de P vers A.

Envoi de la requête ICMP par P à R-eth0. Le routeur reçoit la trame, décapsule, récupère l'@IPdst dans l'en-tête IP=10.189.12.27, cherche dans sa table de routage comment atteindre cette destination : il faut passer par l'interface R-eth1 car le routeur est directement connecté au réseau de la destination. Mais l'@MACdst de 10.189.12.27 est inconnue (?) donc sur le routeur

Fabrication d'une requête ARP encapsulée dans Ethernet pour trouver l'@MAC de 10.189.12.27 (A-eth0) : @MACdst=FF:FF:FF:FF:FF:FF @MACsrc=R-eth1

Envoi de la requête ARP par R. Fabrication et envoi de la réponse ARP par A : @MACdst=R-eth1 @MACsrc=A-eth0 type=0806 (?) est maintenant connu.

Envoi de la requête ICMP par R. @MACsrc=R-eth1 @MACdst=A-eth0

Envoi de la réponse ICMP par A. On suppose ici que A connaît @MAC de R-eth1

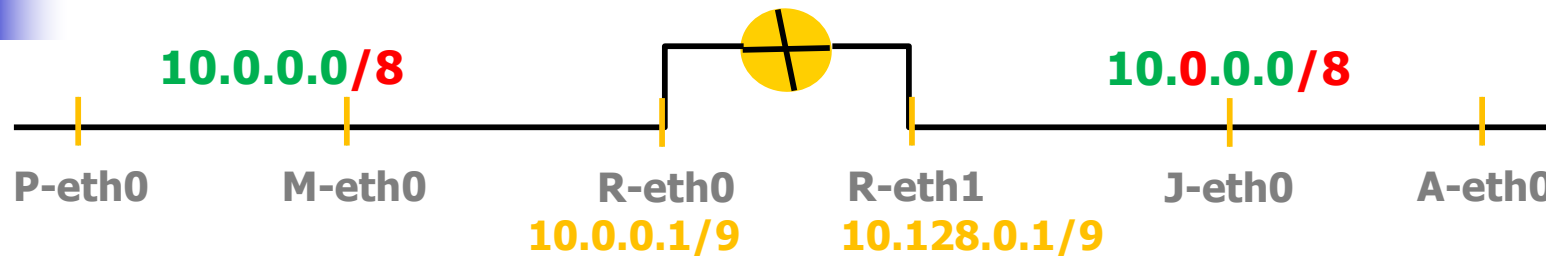
Envoi de la réponse ICMP par R. On suppose ici que R connaît @MAC de P-eth0

Il faut donc 8 trames pour faire un ping-pong entre P et A :

Pour le ping : 2 ARP + 1 ICMP + 2 ARP + 1 ICMP

Pour le pong : 1 ICMP + 1 ICMP

### 3. Proxy ARP



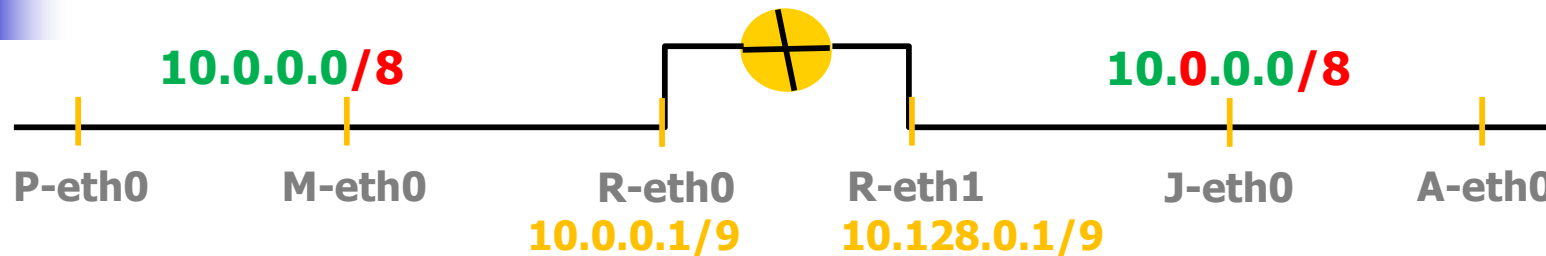
- Cet équipement est maintenant configuré en proxy ARP. Indiquez le contenu du cache ARP de Pierre si toutes les stations ont déjà communiqué entre elles.

Un proxy ARP répond aux requêtes ARP à la place du destinataire. Cela permet de simplifier la configuration des machines en leur faisant croire qu'elles sont toutes dans le même réseau alors que ce n'est pas le cas.

Ici, on change le netmask de P, M, J et A en mettant /8 au lieu de /9. Le routeur garde en revanche /9 sur ses deux interfaces. Donc P, M, J et A pensent qu'elles sont directement connectées à 10.0.0.0/8 et donc qu'elles peuvent joindre les autres stations sans passer par un routeur.

Pour le ping de P vers M, rien ne change. Pour le ping de P vers A, P fabrique et envoie la requête ARP pour trouver l'@MAC de A. Le routeur reçoit la requête et répond à la place de A en donnant son @MAC à lui, celle de R-eth0. Donc cela est équivalent à la situation précédente (sans ProxyARP) dans laquelle la requête ARP cherchait l'@MAC de R-eth0 !

### 3. Proxy ARP



Après tous les ping, le cache ARP de Pierre contient :

@IP	@MAC
celle de M	celle de M-eth0
10.0.0.1	celle de R-eth0
10.128.0.1	celle de R-eth0
celle de J	celle de R-eth0
celle de A	celle de R-eth0

- L'entreprise envisage de raccorder son site à Internet. Dans quelle mesure cela est-il possible ?

Il faut relier le routeur à Internet, configurer le NAT car les machines utilisent des adresses privées et ajouter la route par défaut (0.0.0.0/0) sur toutes les machines.



## 5. Assignation de blocs d'adresses IP

---

- De nombreuses adresses de classe C sont disponibles à partir de l'adresse 194.24.0.0
- 4 organisations A, B, C et D demandent l'une après l'autre des adresses, respectivement 2000 (pour A), 4000 (B), 1000 (C), 2000 (D)
- Comment leur assignez vous les adresses demandées ? Indiquez pour chaque plage la première et la dernière adresse de la plage ainsi que le masque avec la notation t.u.v.w/prefix

## 5. Assignation de blocs d'adresses IP

- 2000 (pour A), 4000 (B), 1000 (C), 2000 (D)

Pour faire 2000 adresses, il faut 11 bits car  $2^{10} \leq 2000 \leq 2^{11}$

Le masque correspondant est donc  $/32-11=/21$ . Donc si A prend les 1ères @ de la plage, SRA=194.24.0.0/21. Avec ce masque, les SR sont les multiples de 8 dans le 3<sup>ème</sup> octet :

$$\begin{array}{ccccccc} \text{8 bits} & + & \text{8 bits} & + & \text{5 bits pour le réseau} & | & \text{11 bits} \\ /21 = & 255 & . & 255 & . & 2^7 & 2^6 & 2^5 & 2^4 & \boxed{2^3} & 2^2 & 2^1 & 2^0 & . & 0 \end{array}$$

Donc la dernière machine dans SRA est 194.24.7.254

Pour faire 4000 adresses, il faut 12 bits car  $2^{11} \leq 4000 \leq 2^{12}$

Le masque correspondant est donc  $/32-12=/20$ . Avec ce masque, les SR sont les multiples de 16 dans le 3<sup>ème</sup> octet. Si on prend la suite des @, SRB=194.24.8.0/20 mais cela n'est pas possible car 8 n'est pas un multiple de 16 donc SRB n'est pas l'adresse d'un SR

Pour donner 4000 adresses en un seul bloc à B, il faut prendre SRB=194.24.16.0/20, @dernière machine = 194.24.31.254

## 5. Assignation de blocs d'adresses IP

- 2000 (pour A), 4000 (B), 1000 (C), 2000 (D)

Une solution possible :

<u>Orga</u>	<u>@SR</u>	<u>@SRsuivant</u>	<u>@1<sup>ère</sup></u>	<u>@dernière</u>	<u>Nb stations</u>
A	194.24.0.0/21	8.0/21	0.1	7.254	2046
B	194.24.16.0/20	32.0/20	16.1	31.254	4094
C	194.24.32.0/22	36.0/22	32.1	35.254	1022
D	194.24.8.0/21	16.0/21	8.1	15.254	2046

Ci-dessus, on a choisi de mettre D dans le trou de taille /21 laissé juste après A. On aurait pu y mettre C avec 8.0/22 mais cela aurait laissé un trou pour la plage 12.0/22. Pour ne pas laisser de trou, il faut attribuer les SR du plus grand au plus petit :

<u>Orga</u>	<u>@SR</u>	<u>@SRsuivant</u>	<u>@1<sup>ère</sup></u>	<u>@dernière</u>	<u>Nb stations</u>
B	194.24.0.0/20	16.0/20	0.1	15.254	4094
A	194.24.16.0/21	24.0/21	16.1	23.254	2046
D	194.24.24.0/21	32.0/21	24.1	31.254	2046
C	194.24.32.0/22	36.0/22	32.1	35.254	1022

## 6. Agrégation de routes

- Un routeur vient de recevoir les routes suivantes :  
57.6.96.0/21, 57.6.104.0/21, 57.6.112.0/20, 57.6.120.0/22
- Si elles utilisent toutes la même ligne de sortie, peuvent-elles être agrégées ? Si oui, comment ? Sinon, pourquoi ?

Pour chaque SR, il faut déterminer le SR suivant.

$$\begin{array}{ccccccc} \text{8 bits} & + & \text{8 bits} & + & \text{5 bits pour le réseau} & | & \text{11 bits} \\ /21 = & 255 & . & 255 & . & 2^7 & 2^6 & 2^5 & 2^4 & \boxed{2^3} & 2^2 & 2^1 & 2^0 & . & 0 \end{array}$$

En /21, les SR sont les multiples de 8 dans le 3<sup>ème</sup> octet.

En /20, les SR sont les multiples de 16 dans le 3<sup>ème</sup> octet.

En /22, les SR sont les multiples de 4 dans le 3<sup>ème</sup> octet.

Le SR suivant de 96.0/21 est 104/21, celui de 104.0/21 est 112.0/21, celui de 112.0/20 est 128.0/20, celui de 120.0/22 est 124.0/22.



## 6. Agrégation de routes

- 57.6.96.0/21, 57.6.104.0/21, 57.6.112.0/20, 57.6.120.0/22

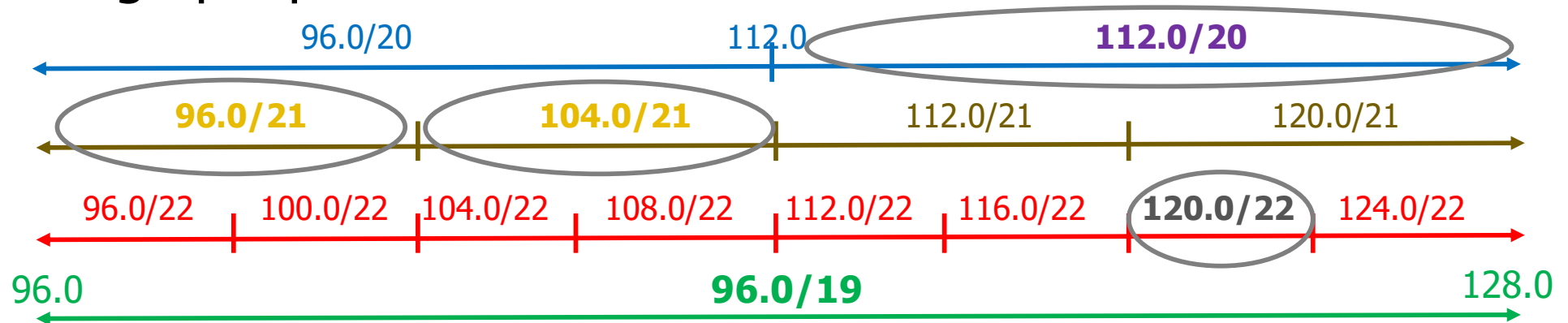
Le SR suivant de 96.0/21 est 104.0/21

celui de 104.0/21 est 112.0/21

celui de 112.0/20 est 128.0/20

celui de 120.0/22 est 124.0/22

Vision graphique des 4 sous-réseaux :



Conclusion : on peut les agréger avec 57.6.96.0/19 qui est un SR car 96 est un multiple de 32 donc compatible avec /19



## 7. Changement de destination

---

- La plage d'adresses IP de 29.18.0.0 à 29.18.127.255 a été agrégée en 29.18.0.0/17 dans une table de routage
- Le bloc de 1024 adresses de 29.18.0.0 à 29.18.3.255 vient d'être affecté à un autre site et doit donc être routé vers une ligne de sortie différente
- Questions :
  - Comment la table de routage doit-elle être modifiée ?
  - Comment un paquet à destination de 29.18.47.38 sera t-il routé ? Idem pour 29.18.1.38 ?

## 7. Changement de destination

- La plage d'adresses IP de 29.18.0.0 à 29.18.127.255 a été agrégée en 29.18.0.0/17 dans une table de routage

$$\text{/17} = \overset{\text{8 bits}}{255} \cdot \overset{\text{8 bits}}{255} \cdot \overset{\text{1 bit}}{2^7} \overset{\text{15 bits pour les machines}}{2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0} \cdot 0$$

Donc en /17, les SR sont les multiples de 128 dans le 3<sup>ème</sup> octet  
29.18.127.255 est la diffusion dans 29.18.0.0/17

- Le bloc de 1024 adresses de 29.18.0.0 à 29.18.3.255 vient d'être affecté à un autre site et doit donc être routé vers une ligne de sortie différente

On cherche le masque du SR 29.18.0.0/? dont la diffusion est 29.18.3.255 donc @SRsuivant=29.18.4.0/? donc le ? pour lequel les SR sont les multiples de  $4=2^2$  donc  $? = 8+8+6 = /22$

$$\text{/?} = 255 \cdot 255 \cdot 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0 \cdot 0$$



## 7. Changement de destination

- Comment la table de routage doit-elle être modifiée ?

Les deux SR sont 29.18.0.0/17 et 29.18.0.0/22

29.18.0.0/22 est inclut dans 29.18.0.0/17 mais a un prochain saut différent donc il faut ajouter une ligne pour ce SR. Par exemple

Destination/masque	Prochain saut ou Interface
29.18.0.0/22	eth0
29.18.0.0/17	29.18.3.254

- Comment un paquet à destination de 29.18.47.38 sera t-il routé ? Idem pour 29.18.1.38 ?

29.18.47.38 est dans 29.18.0.0/17 mais pas dans 29.18.0.0/22 donc il est routé vers 29.18.3.254

29.18.1.38 est dans 29.18.0.0/17 ET dans 29.18.0.0/22 donc il est routé vers eth0 car la ligne la plus précise est choisie. La table de routage est triée de la destination la plus petite à la plus grande.



## 8. Routage CIDR

---

- Un routeur possède les entrées suivantes dans sa table de routage :

<u>Adresse/Masque</u>	<u>Next Hop</u>
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Routeur 1
Par défaut	Routeur 2

- Que fait le routeur lorsqu'il reçoit un paquet à destination des adresses suivantes ?
  - 1) 135.46.63.10      2) 135.46.57.14      3) 135.46.52.2
  - 4) 192.53.41.7      5) 192.53.42.7



## 8. Routage CIDR

---

<u>Adresse/Masque</u>	<u>Next Hop</u>
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Routeur 1
Par défaut	Routeur 2

- Que fait le routeur lorsqu'il reçoit un paquet à destination des adresses suivantes ?

1) 135.46.63.10    2) 135.46.57.14    3) 135.46.52.2    4) 192.53.41.7  
5) 192.53.42.7

Pour chaque adresse IP destination, il faut chercher quelle(s) ligne(s) « match » dans la table de routage c'est-à-dire celles qui contiennent cette adresse dans leur destination. Si plusieurs lignes correspondent, la ligne retenue est la ligne la plus précise c'est à dire celle de la plus petite destination.

## 8. Routage CIDR

$$/22 = \overbrace{255 \cdot 255 \cdot 2^7 2^6 2^5 2^4 2^3}^{8 \text{ bits} + 8 \text{ bits} + 6 \text{ bits pour le réseau}} \boxed{2^2} \overbrace{2^1 2^0 \cdot 0}^{10 \text{ bits}}$$

<u>Adresse/Masque</u>	<u>Next Hop</u>
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
192.53.40.0/23	Routeur 1
Par défaut	Routeur 2

Donc pour chaque destination (ligne), il faut déterminer la dernière adresse de la plage ou l'@SRsuivant : en /22, les SR sont les multiples de 4 dans le 3<sup>ème</sup> octet ; en /23, les multiples de 2 donc le suivant de 135.46.56.0/22 est 60.0/22, le suivant de 135.46.60.0/22 est 64.0/22, le suivant de 192.53.40.0/23 est 42.0/23. La route par défaut inclut toutes les adresses donc elle « match » toujours mais est choisie en dernier recourt car c'est la destination la moins précise (car la plus grande).

## 8. Routage CIDR

$$\text{/22} = \overset{8 \text{ bits}}{255} \cdot \overset{8 \text{ bits}}{255} \cdot \overset{6 \text{ bits pour le réseau}}{2^7 2^6 2^5 2^4 2^3 \boxed{2^2}} \overset{10 \text{ bits}}{2^1 2^0 \cdot 0}$$

Adresse/Masque

Next Hop

135.46.56.0/22

Interface 0

135.46.60.0/22

Interface 1

192.53.40.0/23

Routeur 1

Par défaut

Routeur 2

- 1) 135.46.63.10    2) 135.46.57.14    3) 135.46.52.2    4) 192.53.41.7  
5) 192.53.42.7

135.46.56.0/22 @suivant=60.0/22 donc inclut seulement 2)

135.46.60.0/22 @suivant=64.0/22 donc inclut seulement 1)

192.53.40.0/23 @suivant=42.0/23 donc inclut seulement 4)

0.0.0.0/0 inclut toutes les adresses donc 1) 2) 3) 4) 5)

Donc 1) passe par Interface 1, 2) par Interface 0, 4) par Routeur 1, 3) et 5) par Routeur 2