

Réseaux IP

E. Jeandel

Emmanuel.Jeandel at univ-lorraine.fr

- Attribuer une adresse à une machine
- Savoir comment envoyer un message à une machine connaissant uniquement son adresse
- Trouver à partir de l'adresse, dans quel réseau se trouve la machine

Idée de base : Pour envoyer un message à une machine d'adresse A

- Soit je m'aperçois que A est dans mon réseau local, et je lui envoie directement
- Soit ce n'est pas le cas, et je trouve quelqu'un (un *routeur*) pour le faire à ma place.

La partie routage sera faite la semaine prochaine.

Les adresses liaisons (MAC) ne suffisent pas :

- L'adresse liaison identifie uniquement la machine (\sim numéro de série), mais pas où elle se situe
- L'adresse doit permettre de savoir comment accéder à la machine
- On peut vouloir changer la carte réseau/la machine sans que ça ne se voie.
- On peut vouloir changer l'adresse

- Chaque machine est dotée d'une adresse A
- Chaque machine possède un couple (A, B)
 - A : adresse de la machine.
 - B : ensemble des adresses situées sur le même réseau.

Quand on parle de machine, il faut plutôt parler d'accès réseau : un routeur aura plusieurs adresses.

1 IPv4

2 Réseau local

IPv4 est la quatrième édition de IP (les 3 premières n'ont pas existé plus de six mois). C'est à la fois :

- Une façon d'adresser des machines
- Un protocole pour communiquer entre ces machines

Adresses IPv4

- Une adresse IPv4 est la donnée de 32 bits (4 octets).
- On la représente souvent sous la forme de 4 blocs de 8 bits, chaque bloc représenté sous la forme d'un entier

- Exemple

bin	01110110	.	00010011	.	01011110	.	11110101
dec	118	.	19	.	94	.	245
hex	76	.	13	.	5E	.	F5

Comment trouver l'ensemble des machines sur le même réseau ?

Préfixes (CIDR)

Historiquement : réseaux de 256, 65536 ou 16777216 adresses : Le premier octet permet de savoir la taille du réseau.

Maintenant, adressage par **préfixe** :

- Préfixe de taille n : les machines dans le réseau sont toutes les machines qui ont les mêmes n premiers bits. On écrit l'adresse sous la forme $xxx.yyy.zzz.www/n$
- Peut être donné par un *masque*, formé par n bits à 1 et $32 - n$ bits à 0.

Exemple

118.19.94.245/21 ?

dec	118	.	19	.	94	.	245
bin	01110110	.	00010011	.	01011110	.	11110101

Exemple

118.19.94.245/21 ?

dec	118	.	19	.	94	.	245
bin	01110110	.	00010011	.	01011110	.	11110101

On isole les 21 premiers bits

Exemple

118.19.94.245/21 ?

dec	118	.	19	.	94	.	245
bin	01110110	.	00010011	.	01011110	.	11110101

Le réseau est constitué de toutes les machines qui ont les mêmes 21 premiers bits

Exemple

118.19.94.245/21 ?

dec	118	.	19	.	94	.	245
bin	01110110	.	00010011	.	01011110	.	11110101

De

dec	118	.	19	.	88	.	0
bin	01110110	.	00010011	.	01011000	.	00000000

à

dec	118	.	19	.	95	.	255
bin	01110110	.	00010011	.	01011111	.	11111111

Exemple

118.19.94.245/21 ?

dec	118	.	19	.	94	.	245
bin	01110110	.	00010011	.	01011110	.	11110101

Masque de réseau :

dec	255	.	255	.	248	.	0
bin	11111111	.	11111111	.	11111000	.	00000000

Exemple (TODO)

125.13.58.142/22 ?

Exemple (TODO)

125.13.58.142/22 ?

dec	125	.	13	.	58	.	142
bin	01111101	.	00001101	.	00111010	.	10001110

On isole les 22 premiers bits

Exemple (TODO)

125.13.58.142/22 ?

dec	125	.	13	.	58	.	142
bin	01111101	.	00001101	.	00111010	.	10001110

Le réseau est constitué de toutes les machines qui ont les mêmes 22 premiers bits

Exemple (TODO)

125.13.58.142/22 ?

dec	125	.	13	.	58	.	142
bin	01111101	.	00001101	.	00111010	.	10001110

De

dec	125	.	13	.	56	.	0
bin	01111101	.	00001101	.	00111000	.	00000000

à

dec	125	.	13	.	59	.	255
bin	01111101	.	00001101	.	00111011	.	11111111

Exemple (TODO)

125.13.58.142/22 ?

dec	125	.	13	.	58	.	142
bin	01111101	.	00001101	.	001110	10	10001110

Masque de réseau :

dec	255	.	255	.	252	.	0
bin	11111111	.	11111111	.	11111100	.	00000000

Exemple (TODO)

```
mamachine$ ifconfig eth0  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500  
inet 140.117.13.101 netmask 255.255.248.0 broadcast 140.117.15.255
```

Exemple (TODO)

```
mamachine$ ifconfig eth0  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500  
inet 140.117.13.101 netmask 255.255.248.0 broadcast 140.117.15.255
```

dec	140	.	117	.	13	.	101
bin	10001100	.	01110101	.	00001101	.	01100101

dec	255	.	255	.	248	.	0
bin	11111111	.	11111111	.	11111000	.	00000000

C'est un préfixe /21

Exemple (TODO)

```
mamachine$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
inet 140.117.13.101 netmask 255.255.248.0 broadcast 140.117.15.255
```

dec		140	.	117	.	13	.	101
bin		10001100	.	01110101	.	00001101	.	01100101

Le réseau est constitué de toutes les machines qui ont les mêmes 21 premiers bits

Exemple (TODO)

```
mamachine$ ifconfig eth0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
inet 140.117.13.101 netmask 255.255.248.0 broadcast 140.117.15.255
```

dec	140	.	117	.	13	.	101
bin	10001100	.	01110101	.	00001101	.	01100101

De

dec	140	.	117	.	8	.	0
bin	10001100	.	01110101	.	00001000	.	00000000

à

dec	140	.	117	.	15	.	255
bin	10001100	.	01110101	.	00001111	.	11111111

Adresses particulières dans un réseau

Deux adresses particulières ne sont pas utilisables pour une interface réseau

- La première adresse, qui est l'adresse du *réseau*
- La dernière adresse, qui est l'adresse de *broadcast*. Envoyer à cet adresse permet d'envoyer à toutes les machines du réseau.

Exemple : 118.19.94.245/21

- Adresse réseau : 118.19.88.0
- Adresse broadcast : 118.19.95.255

Toutes les autres adresses peuvent être utilisées pour une interface réseau.

Note : un réseau /31 ou /32 ne contient aucune adresse pour une interface réseau.

Préfixes importants

Adresses `xxx.yyy.zzz.www/24` :

- Toutes les adresses entre `xxx.yyy.zzz.0` et `xxx.yyy.zzz.255`
- 256 adresses en tout
- Adresse du réseau : `xxx.yyy.zzz.0/24`
- Adresse de broadcast : `xxx.yyy.zzz.255`
- 254 adresses utilisables par les interfaces réseaux.

Préfixes importants

Adresses `xxx.yyy.zzz.www/16` :

- Toutes les adresses entre `xxx.yyy.0.0` et `xxx.yyy.255.255`
- 65536 adresses en tout
- Adresse du réseau : `xxx.yyy.0.0/16`
- Adresse de broadcast : `xxx.yyy.255.255`
- 65534 adresses utilisables par les interfaces réseaux.

Préfixes importants

Adresses 0.0.0.0/0 :

- Toutes les adresses possibles
- Désigne tout le réseau internet

Adresses xxx.yyy.zzz.www/32 :

- Désigne une seule adresse
- Pas vraiment un réseau

Un préfixe $/n$ contient 2^{32-n} adresses, dont 2 ne sont pas utilisables

- $n = 24$, $2^8 = 256$ adresses, dont 254 utilisables pour une interface réseau
- $n = 20$, $2^{12} = 4096$ adresses, dont 4094 utilisables pour une interface réseau

Réseaux privés (RFC 8190)

Réseau privé : Adresse **pas unique** sur l'Internet, et sont inconnus des routeurs par défaut

- 0.0.0.0/8 : peut désigner le réseau local (uniquement comme adresse source)
- 127.0.0.0/8 : boucle locale : ne contient que la machine
- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16
- 100.64.0.0/10 conçu pour connecter des réseaux privés ensemble

Comment fixer l'adresse IP d'une machine ?

Deux méthodes

- Statique : dans un fichier de configuration sur la machine
- Dynamique : obtenue au démarrage de la machine, peut changer d'un jour à l'autre (DHCP, voir cours de L3)

Attention : il faut fixer l'adresse IP mais aussi le masque(préfixe) !

1 IPv4

2 Réseau local

Idée de base : Pour envoyer un message à une machine d'adresse A

- Soit je m'aperçois que A est dans mon réseau local, et je lui envoie directement
- Soit ce n'est pas le cas, et je trouve quelqu'un (un *routeur*) pour le faire à ma place.

Soit la machine 12.14.59.39/24. Parmi ces adresses IP, lesquelles sont sur le réseau local ?

- 12.14.59.1
- 12.14.16.64
- 12.14.59.17
- 6.6.6.6
- 12.14.59.39

Sur un réseau local, on communique en utilisant les adresses MAC.

Il faut un mécanisme pour trouver l'adresse MAC à partir de l'adresse IP.

Ce mécanisme (protocole) s'appelle ARP.

ARP

Address Resolution Protocol (RFC826)

Un paquet ARP contient :

- Un champ demande/réponse
- L'adresse IP(resp. MAC) de l'émetteur
- L'adresse IP(resp. MAC) du destinataire

ARP est conçu pour fonctionner avec n'importe quel protocole liaison/réseau

- s/IP/réseau/
- s/MAC/liaison/

ARP

Principe (1/2)

Si A d'adresse IP 1.6.6.4 et d'adresse MAC 00:de:ad:be:ef:00 veut connaître l'adresse MAC de la machine d'adresse IP 1.6.6.15

- Trame ethernet en broadcast (sur ff:ff:ff:ff:ff:ff)
- ARP en mode demande
- Adresse IP/MAC de l'expéditeur (pourquoi ?)
- Adresse IP du destinataire
- Adresse MAC du destinaire : 00:00:00:00:00:00

ARP

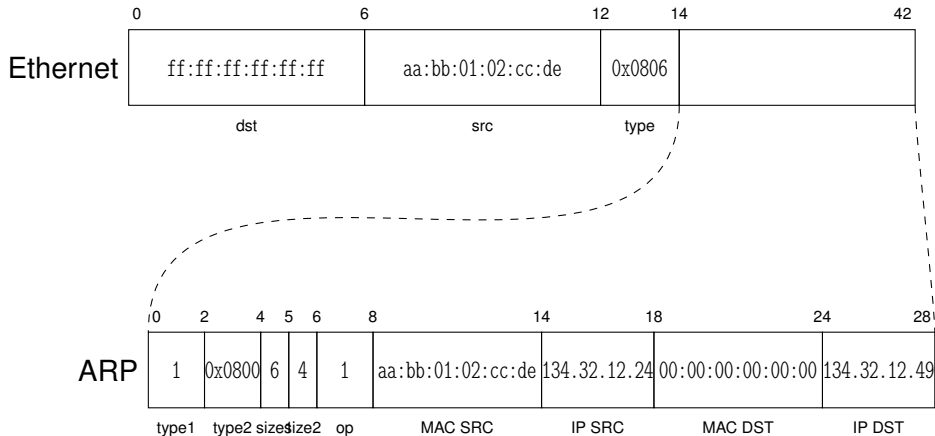
Principe (2/2)

Lorsque *B* veut répondre :

- Trame ethernet *généralement* vers *A*
- ARP en mode réponse
- Adresse IP/MAC de *A* et de *B*

Attention : seul *B* doit répondre.

Exemple pratique



Exemple pratique

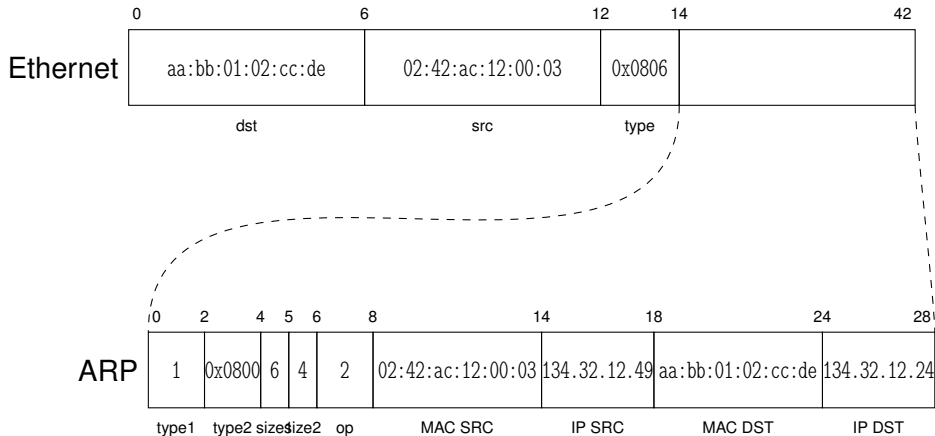


Table ARP

On ne redemande pas à chaque envoi les correspondances IP-Mac !
Cache ARP dans chaque machine :

Adresse	TypeMap	AdresseMat	Indicateurs	Iface
192.168.1.1	ether	a4:2b:b0:f6:a0:d6	C	eth0
192.168.1.4	ether	28:39:26:23:8b:7d	C	eth0
192.168.1.6	ether	48:b0:2d:2d:c7:9a	C	eth0
192.168.1.9	ether	00:11:32:93:06:79	C	eth0
192.168.1.15	ether	b8:27:eb:e4:36:82	CM	eth0
192.168.1.16		(incomplete)		eth0

Indicateurs :

- C : Apprise
- CM : codée en dur
- incomplete : la correspondance a échoué

Mis à jour à quatre moments :

- Quand on reçoit une réponse à une requête ARP
- Quand on reçoit une question nous concernant
- Quand on reçoit un paquet ARP sur une adresse IP *qu'on connaît* :
- Au bout d'un certain temps, on vide les vieilles entrées (OS dependent).

Mis à jour à quatre moments :

- Quand on reçoit une réponse à une requête ARP
 - J'ai demandé qui a l'adresse IP 192.168.0.1
 - On m'a répondu "c'est aa:bb:cc:dd:ee:ff"
 - Je mets ça dans mon cache
- Quand on reçoit une question nous concernant
- Quand on reçoit un paquet ARP sur une adresse IP *qu'on connaît* :
- Au bout d'un certain temps, on vide les vieilles entrées (OS dependent).

Mis à jour à quatre moments :

- Quand on reçoit une réponse à une requête ARP
- Quand on reçoit une question nous concernant
 - J'ai vu passé la demande "qui a l'adresse IP 192.168.0.1", qui est mon adresse IP, venant de la machine 192.168.0.4 dont l'adresse mac est 01:02:03:04:05:06
 - Je réponds, parce que je suis civilisé
 - Je mets dans mon cache que 192.168.0.4 correspond à 01:02:03:04:05:06.
 - Si elle m'a contacté, je vais sans doute répondre plus tard
- Quand on reçoit un paquet ARP sur une adresse IP *qu'on connaît* :
- Au bout d'un certain temps, on vide les vieilles entrées (OS dependent).

Mis à jour à quatre moments :

- Quand on reçoit une réponse à une requête ARP
- Quand on reçoit une question nous concernant
- Quand on reçoit un paquet ARP sur une adresse IP *qu'on connaît* :
 - J'ai vu passé la question "qui a l'adresse IP 192.168.0.1 ? C'est 192.168.0.4, d'adresse 01:02:03:04:05:06 qui demande" alors que je n'ai rien demandé
 - Je pensais que 192.168.0.4 avait comme adresse MAC de:ad:be:ef:00:00, on m'aurait donc menti
 - Je change donc dans mon cache
 - Si je ne connaissais pas 192.168.0.4, je ne fais rien
- Au bout d'un certain temps, on vide les vieilles entrées (OS dependent).

Mis à jour à quatre moments :

- Quand on reçoit une réponse à une requête ARP
- Quand on reçoit une question nous concernant
- Quand on reçoit un paquet ARP sur une adresse IP *qu'on connaît* :
- Au bout d'un certain temps, on vide les vieilles entrées (OS dependent).

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66



IP	MAC

IP	MAC



192.168.1.3/24
aa:bb:cc:dd:ee:ff

A veut envoyer un message à 192.168.1.4

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66



IP	MAC

IP	MAC



192.168.1.3/24
aa:bb:cc:dd:ee:ff

1. A regarde si 192.168.1.4 est dans son réseau local \Rightarrow Oui.

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66



IP	MAC

IP	MAC



192.168.1.3/24
aa:bb:cc:dd:ee:ff

2. A consulte son cache ARP

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66



IP	MAC

IP	MAC



192.168.1.3/24
aa:bb:cc:dd:ee:ff

3. A ne connait pas encore l'adresse MAC de 192.168.1.4

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66



IP	MAC

IP	MAC



192.168.1.3/24
aa:bb:cc:dd:ee:ff

4. A envoie une requête ARP sur son réseau local.

Entête Ethernet source : aa:bb:cc:dd:ee:ff

destination : ff:ff:ff:ff:ff:ff (broadcast)

Entête ARP

source : 192.168.1.3/aa:bb:cc:dd:ee:ff

destination : 192.168.1.4/00:00:00:00:00:00

type : question

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66



IP	MAC

IP	MAC



192.168.1.3/24
aa:bb:cc:dd:ee:ff

5. B et C reçoivent la trame. B s'en félicite et passe à autre chose.

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66

IP	MAC



C

IP	MAC



192.168.1.3/24
aa:bb:cc:dd:ee:ff

6. C ajoute la correspondance ARP à sa table

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.3/24
aa:bb:cc:dd:ee:ff

IP	MAC

192.168.1.4/24
ff:66:66:66:66:66



C

IP	MAC
192.168.1.3	aa:bb:cc:dd:ee:ff

6. C ajoute la correspondance ARP à sa table

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66



IP	MAC
192.168.1.3	aa:bb:cc:dd:ee:ff

IP	MAC



192.168.1.3/24
aa:bb:cc:dd:ee:ff

7. C répond à A.

Entête Ethernet

source : ff:66:66:66:66:66

destination : aa:bb:cc:dd:ee:ff

Entête ARP

source : 192.168.1.4/ff:66:66:66:66:66

destination : 192.168.1.3/aa:bb:cc:dd:ee:ff

type : réponse

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66



IP	MAC
192.168.1.3	aa:bb:cc:dd:ee:ff

IP	MAC



192.168.1.3/24
aa:bb:cc:dd:ee:ff

8. A ajoute la correspondance ARP à sa table

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



B

192.168.1.4/24
ff:66:66:66:66:66



IP	MAC
192.168.1.3	aa:bb:cc:dd:ee:ff

C

IP	MAC
192.168.1.4	ff:66:66:66:66:66



A

192.168.1.3/24
aa:bb:cc:dd:ee:ff

8. A ajoute la correspondance ARP à sa table

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66



IP	MAC
192.168.1.3	aa:bb:cc:dd:ee:ff

IP	MAC
192.168.1.4	ff:66:66:66:66:66



192.168.1.3/24
aa:bb:cc:dd:ee:ff

9. A a toutes les infos pour envoyer son paquet IP.

Entête Ethernet source : aa:bb:cc:dd:ee:ff
 destination : ff:66:66:66:66:66

Entête IP source : 192.168.1.3
 destination : 192.168.1.4

message bonjour !

Exemple

192.168.1.6/24
00:11:22:33:44:55

IP	MAC



192.168.1.4/24
ff:66:66:66:66:66



IP	MAC
192.168.1.3	aa:bb:cc:dd:ee:ff

IP	MAC
192.168.1.4	ff:66:66:66:66:66



192.168.1.3/24
aa:bb:cc:dd:ee:ff

10. Si jamais C veut répondre à A, il n'a pas besoin de refaire une requête ARP.

Ce qu'il faut retenir

Avant tout envoi d'un ou plusieurs paquets IP, on voit en général un échange ARP

Sauf si la correspondance est dans le cache.

Attention, seule la machine qui a l'adresse IP doit répondre à la requête ARP

- Le commutateur ne doit pas essayer d'être malin
- La machine peut avoir changé d'adresse IP
- La machine peut avoir changé d'endroit

Si on change l'adresse IP d'une machine :

- Elle recevra immédiatement les paquets à destination de la nouvelle adresse IP (pourquoi ?)
- Elle recevra aussi certains paquets à destination de l'ancienne adresse IP (pourquoi ?)

Si on change l'adresse MAC d'une machine (parce qu'on a changé la carte réseau)

- Elle ne recevra pas tout de suite les paquets (pourquoi ?)

Que se passe-t-il si deux machines ont la même adresse IP ?

Dans un réseau local, ce n'est pas vraiment l'adresse IP qui décide qui va recevoir le paquet, mais plutôt la correspondance ARP qui a été effectuée

- Possibilité d'attaques : une machine répond à la requête ARP à la place d'une autre (ARP spoofing/cache poisoning)