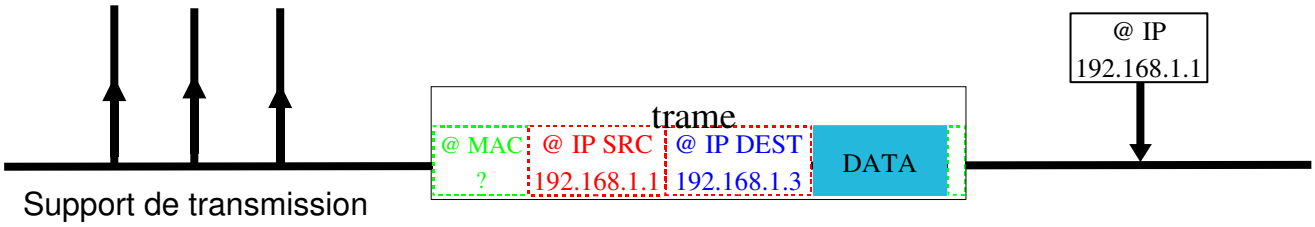
Table des matières

Le protocole ARP (*Address Resolution Protocol*)

Le protocole ARP sert à traduire une adresse réseau IP en une adresse physique.

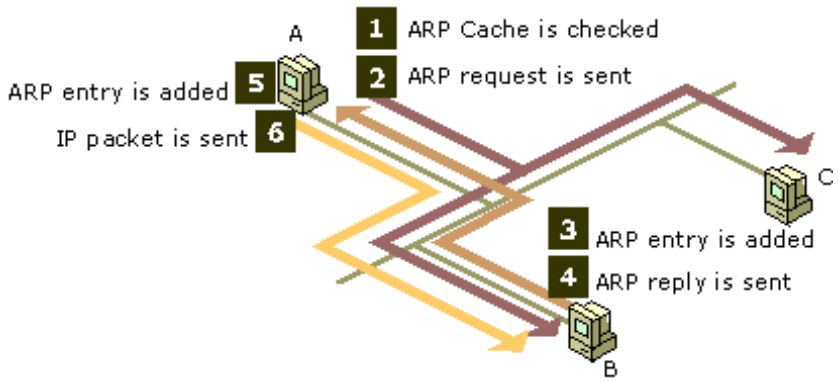
Un poste désire envoyer un paquet IP à un poste appartenant au même réseau physique que lui. Il doit connaître l'adresse physique du destinataire. Or, il ne connait que son adresse IP.



Le protocole ARP va lui permettre de trouver l'adresse physique du poste destinataire. Ce mécanisme est transparent pour l'utilisateur.

Une table de conversion est générée dynamiquement sur chaque hôte dans ce qu'on appelle l'"***ARP cache***". Quand ARP reçoit une demande de conversion, il consulte sa table et retourne l'adresse physique si elle s'y trouve sinon il envoie un paquet spécial *ARP Request Packet* à tous les hôtes du même réseau physique incluant l'adresse IP à rechercher et en utilisant l'adresse broadcast MAC FF FF FF FF FF FF.

La machine possédant l'adresse réseau IP demandée répond en lui renvoyant donc son adresse physique qui est alors placée dans la table ARP.



Si aucune réponse n'est reçue dans un délai imparti, la requête est envoyée à nouveau.

Le contenu de l'*ARP Cache* est généralement conservé jusqu'à l'extinction de la machine hôte. Lors du démarrage de la machine, l'*ARP Cache* est donc vide :

```
# arp -v
Entrées: 0      Ignorées: 0     Trouvées: 0

# cat /proc/net/arp
IP address       HW type     Flags       HW address            Mask     Device
```

Remarques:

- ### *Routeurs et requêtes ARP*

Les requêtes ARP ne passent pas les routeurs, qui relaient des informations au niveau de la couche réseau mais pas du trafic *broadcast* MAC.

- ### *ARP Spoofing (ou ARP Redirect)*

Si une machine non fiable a accès au réseau physique et émet de faux messages ARP pour corrompre le cache ARP d'une machine cible et d'en détourner tout le trafic vers elle-même afin d'en d'écouter et/ou modifier les données (attaque par ARP *spoofing*). Des générateurs de paquets ARP comme **arpspoof** ou **nemesis** permettent ce type d'attaque. La machine pirate se rend transparente en reroutant le trafic en ayant activé l'*IP Forwarding* (`echo 1 > /proc/sys/net/ipv4/ip_forward`). Il existe donc un certain risque lié à l'utilisation du protocole ARP, même sur des réseaux segmentés par des *switch*.

- ### *Proxy ARP*

Le proxy-ARP permet de résoudre le problème posé par des ***firewall*** installés sur un même réseau d'adresse. Les requêtes ARP, étant faites par *broadcast*, seront donc normalement bloquée par la machine filtrante. L'activation de la fonctionnalité Proxy-ARP permet de palier à ce problème en demandant explicitement à ce que les requêtes et réponses ARP arrivant par une carte soient propagées vers l'autre et vice-versa.
Sous Linux, on activera le Proxy-ARP pour chacune des 2 interfaces de la manière suivante :

```
# echo 1 > /proc/sys/net/ipv4/conf/eth0/proxy_arp
# echo 1 > /proc/sys/net/ipv4/conf/eth1/proxy_arp
```

Le protocole RARP (Reverse ARP)

Le protocole RARP permet d'associer une adresse réseau à une adresse physique. Ce protocole était utilisé avant l'adoption du protocole DHCP. RARP était alors utilisé sous Unix par des stations *diskless* (sans disque) ou des terminaux pour leur attribuer une adresse IP à partir d'un serveur. Le protocole *Inverse* ARP est similaire à RARP.

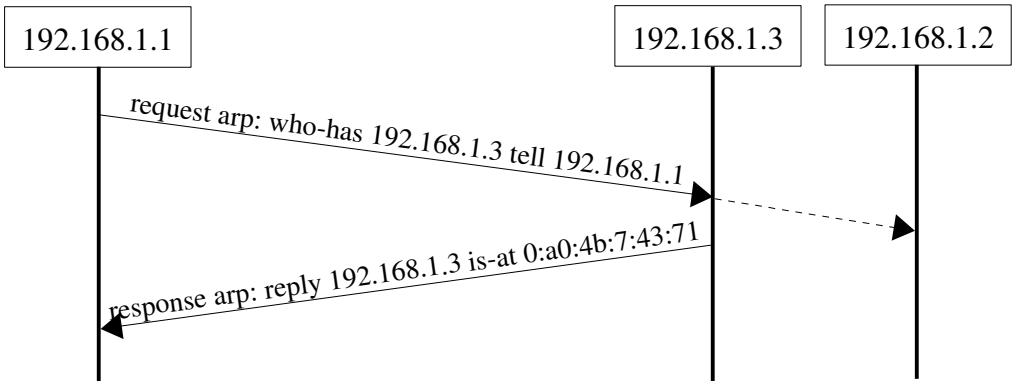
Les RFCs

RFC	***Contenu***
RFC 826	ARP
RFC 903	RARP
RFC 1122	Prérequis réseau
RFC 1433	ARP direct
RFC 1868	Extension UnARP
RFC 2131	DHCP et DHCP ARP
RFC 2390	Inverse ARP

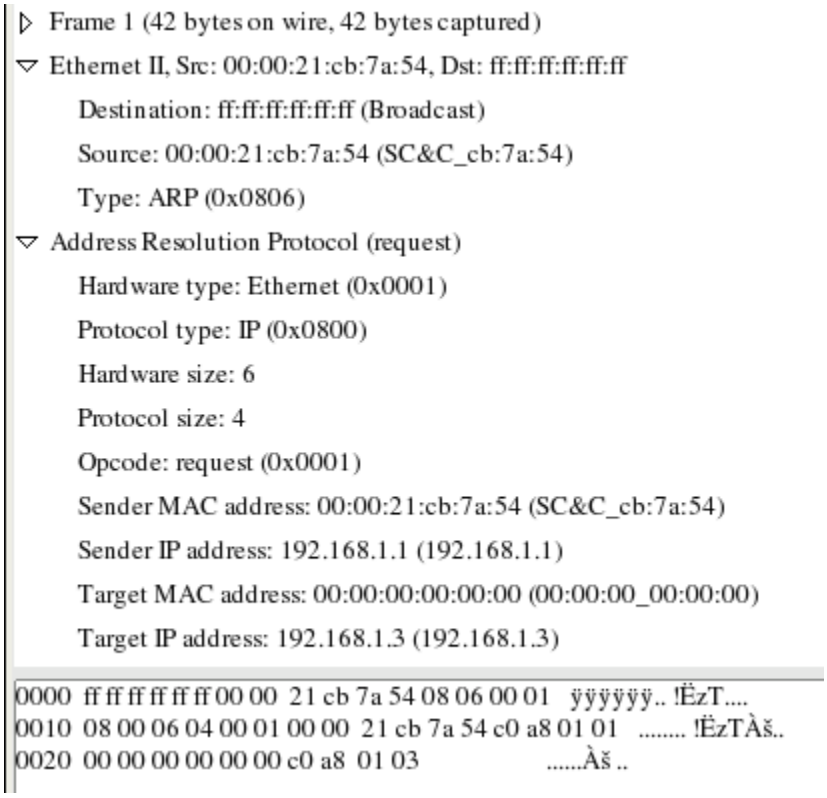
Exemple

On vient de démarrer le poste d'adresse IP 192.168.1.1 et le chache ARP est vide. On fait un ping vers le poste d'adresse IP 192.168.1.3. On capture l'échange ARP qui s'en suit avec **tcpdump** :

```
# tcpdump -vv arp
tcpdump: listening on eth0
08:51:26.702516 arp who-has 192.168.1.3 tell 192.168.1.1
08:51:26.702747 arp reply 192.168.1.3 is-at 0:a0:4b:7:43:71
```



Dans le même échange, on capture et on visualise la première trame avec **ethereal** :



La trame ARP (Address Resolution Protocol)

ARP a été conçu pour fonctionner avec n'importe quel protocole, cela implique l'utilisation d'un format de paquet souple. Par exemple, le "paquet" ARP est encapsulé dans une trame Ethernet_II avec l'adresse de diffusion générale (*broadcast*) et le numéro de protocole désignant un paquet ARP (0x806). Les champs spécifiques à ARP sont :

<table>
<tr><td colspan="3">0 8 16 24 31</td></tr>
<tr><td colspan="2">Type matériel</td><td>Type protocole</td></tr>
<tr><td>lg @ physique</td><td>lg @ IP</td><td>Opération</td></tr>
<tr><td colspan="3">@ physique émetteur …</td></tr>
<tr><td colspan="2">@ physique émetteur.</td><td>@ IP émetteur …</td></tr>
<tr><td colspan="2">@ IP émetteur.</td><td>@ physique récepteur …</td></tr>
<tr><td colspan="3">@ physique récepteur.</td></tr>
<tr><td colspan="3">@ IP récepteur.</td></tr>
</table>

Pour rappel:

- les adresse physiques ont une taille de 6 octets
- les adresses réseaux IP v4 ont une taille de 4 octets

Quelques valeurs du champ **Type matériel** :

Type matériel	**Protocole**
1	**Ethernet**
6	IEEE 802
20	ATM

Le champ **Type protocole** désigne le protocole dont on veut résoudre les adresses : toujours 0x800 pour IP.
Le champ **lg @ physique** indique la longueur de l'adresse physique soit 6 octets pour Ethernet ;
Le champ **lg @ protocole** indique la longueur de l'adresse réseau soit 4 octets pour IPv4 ;

Le champ **Opération** définit la signification du paquet, soit :

Opération	**Message**
1	**Requête ARP**
2	**Réponse ARP**
3	Requête RARP
4	Réponse RARP
8	Requête *Inverse* ARP
9	Réponse *Inverse* ARP

Séquence 1 - ARP

1 . Le cache ARP

1 . Afficher le cache ARP en tapant la commande arp [-n] -a ou en visualisant le fichier /proc/net/arp.
2 . Faire un ping vers un autre poste du réseau.
3 . Que contient le cache ARP après le ping ?
4 . Faire un ping sur une machine située derrière la passerelle, afficher et expliquer le contenu du cache arp. *Remarque:* vérifier qu'il existe une route par défaut vers cette passerelle

2 . La commande ARP

1 . Comment vider le cache arp ?
2 . Faire un ping vers un autre poste du réseau.
3 . Exécuter les commandes suivantes :

```
ifconfig eth0 down
ifconfig eth0 up
arp -n -a
```

4 . En vous aidant de la commande man ifconfig, expliquer ce que font les commandes ifconfig eth0 down et ifconfig eth0 up.
5 . Quel est le contenu du cache après ces deux commandes.
6 . Ajouter manuellement, avec la commande arp -s, une entrée dans le cache ARP pour un poste de votre réseau. Donner la commande exacte.
7 . Afficher le cache arp et indiquer quel est le type de l'entrée. Tester cette entrée avec ping.
8 . Pour conclure, donner les différentes possibilités de la commande arp.

3 . Tromper ARP avec une adresse inexistante

1 . Utiliser la commande arp -s pour modifier l'adresse matérielle de la passerelle par défaut. Donner une fausse adresse, par exemple 08:00:02:22:22:20.
2 . Faire un ping sur la passerelle. Noter et expliquer le résultat obtenu.
3 . Faire un ping sur une machine située derrière la passerelle. Noter et expliquer le résultat obtenu.
4 . Supprimer cette entrée incorrecte. Donner la commande exacte.

4 . Tromper ARP avec une adresse existante mais mal associée

1 . Relever sur le réseau une adresse MAC d'un poste et une adresse IP d'un autre poste.
2 . Affecter avec arp l'adresse IP à l'adresse MAC relevée.
3 . Tester l'adresse IP avec ping.
4 . Noter et expliquer le résultat obtenu.

5 . Analyse de paquets ARP

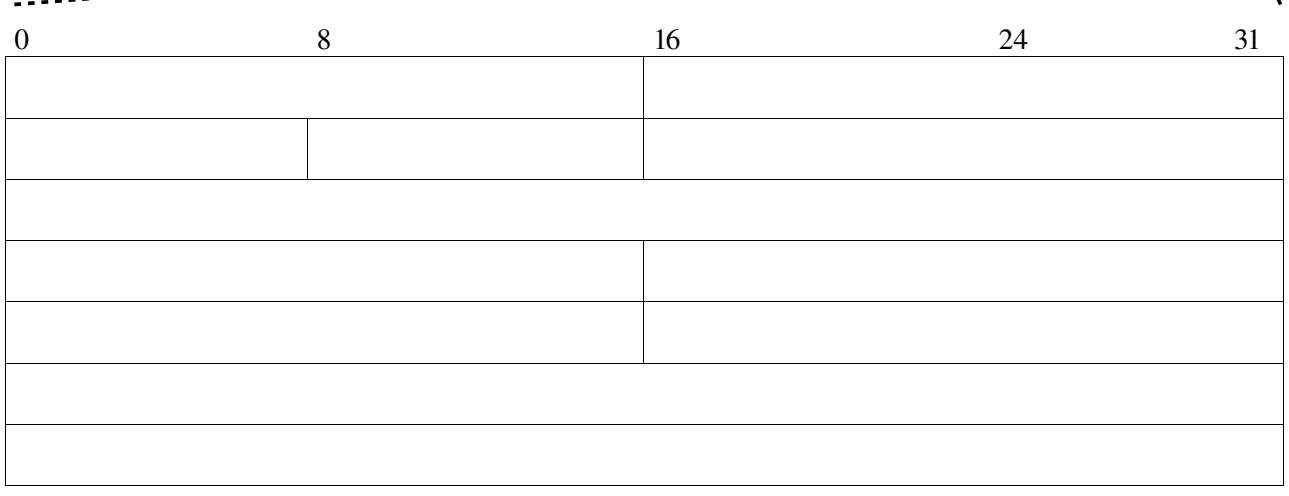
Avec tcpdump:

1 . Démarrer une capture réseau avec la commande tcpdump -e -vv arp.
2 . Commenter les options de la commande tcpdump.
3 . Générer un trafic réseau avec la commande ping sur une adresse inexistante du réseau.
4 . Expliquer le résultat de la capture réalisée.

Avec ethereal:

1 . Démarrer une capture.
2 . Générer un trafic réseau avec la commande ping sur une adresse existante du réseau.
3 . Arrêter la capture
4 . Identifier les deux paquets ARP d'un même échange : quels sont alors le champ Opération (*Opcode*) de ces deux paquets ?
5 . Afficher un paquet *request arp* et donner l'adresse MAC destination. A qui est destiné cette trame ?
6 . Afficher un paquet *reply arp* et réaliser le décodage suivant :

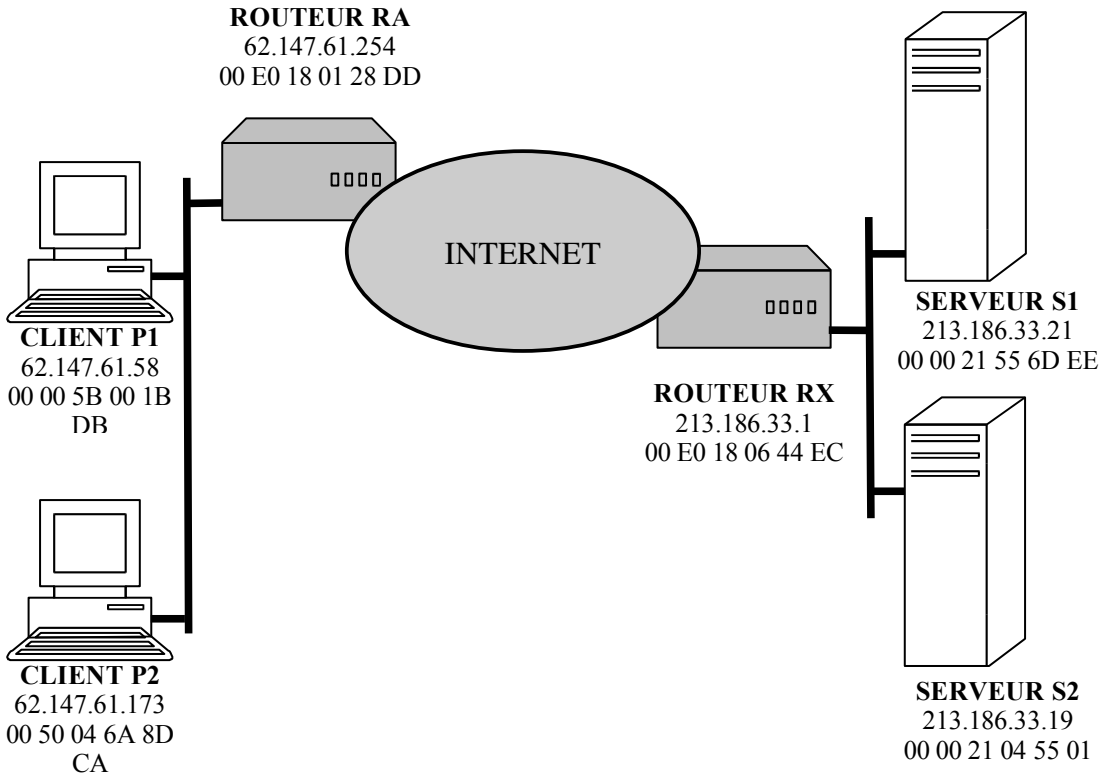
En-tête Ethernet				**Data**
Valeurs hexa				
Signification				



Paquet ARP

6 . Bilan

Le poste CLIENT P2 d'adresse IP 62.147.61.173 fait une requête HTTP vers le serveur WEB d'adresse IP 213.186.33.19. Cette adresse IP n'appartenant pas à son réseau, le client va donc envoyer sa requête vers son routeur INTERNET, dont il connaît l'adresse IP, pour qu'il puisse l'acheminer.



1 . Quelle est l'adresse IP Source contenue dans le paquet envoyé par le client P2 ?

2 . Quelle est l'adresse IP Destination contenue dans le paquet envoyé par le client P2 ?

3 . Comment le client P2 peut-il envoyer ce paquet au routeur INTERNET pour qu'il puisse l'acheminer ?

4 . Représenter sous forme d'un diagramme les échanges (on ne tient pas compte des échanges entre les 2 routeurs)