M2PCCI

Cheick CISSOKO

Gaëtan LAGIER

Eric THIERRY

UNIVERSITÉ Grenoble Alpes

Bien.

CR réseau TP2

2016/2017

LE PROTOCOLE ARP

1) Ce qui permet d'identifier les paquets comme étant de type ARP est le champ "type" de la trame Ethernet. En effet lors de la capture de paquet avec l'application Wireshark, il est indiqué Type : ARP dans la trame Ethernet.

```
> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
v Ethernet II, Src: Broadcom_89:e9:ed (00:10:18:89:e9:ed), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    > Source: Broadcom_89:e9:ed (00:10:18:89:e9:ed)
      Type: ARP (0x0806)
      Padding: 000000000000000000000000000000000000
> Address Resolution Protocol (request)
```

2) Dans la trame Ethernet, le paquet ARP se situe dans le champ Data.

3) Format de la trame ARP :

Hardware Type	Protocol Type	Hardware Size	Protocol Size	Opcode	@ Mac Source	@ IP Source	@ Mac Destination	@ IP Destination
2bytes	2bytes	1byte	1byte	2bytes	6bytes	4bytes	6bytes	4bytes

- Les champs Hardware (type t1 et size s1) et Protocole (type t2 et size s2) représentent respectivement le type t1 de taille s1 du protocole dont on veut la résolution en un autre type t2 de taille s2. Ces champs ne sont plus réellement utilisés aujourd'hui. En effet, le protocole ARP est exclusivement utilisé pour résoudre des adresses Ethernet en adresse IP, mais pour être universel, il a été créé ainsi.
- Le champ Opcode permet de préciser si le paquet est une requête ou une réponse : Request (x01), Reply (x02).
- Les champs restants sont respectivement l'adresse Ethernet et l'adresse IP de la machine qui envoie le paquet, puis l'adresse Ethernet et l'adresse IP de la machine qui est le destinataire de ce paquet.

4) Le niveau Ethernet ne connaît pas la taille des paquets. Il ne peut donc pas en déterminer la fin.

RMQ : Les octets de bourrage correspondent à la parie PADDING située après le protocole ARP.

5) A la réception du paquet, toute la partie data est transmise à la couche ARP par la couche Ethernet. Comme le protocole Ethernet ne connaît pas la taille du paquet, il transmet la partie data + le PADDING qu'il considère comme faisant partie des données. Il n'a donc pas connaissance de ce bourrage.

6) **Algorithme du protocole ARP** :

Tant que (vrai) faire :
Attendre (évènement) :

- **Si** évènement est « Question sur adresse Internet » (requête interne de IP vers ARP), consulter si la table ARP contient l'adresse Ethernet associée à l'adresse IP ciblée. Si la table contient l'adresse, celle ci est transmise au protocole IP, sinon, un paquet ARP request est envoyé sur le réseau pour trouver l'adresse demandée par IP. **Finsi**
- **Si** évènement est expiration du timer d'effacement associé à une entrée, un paquet ARP request est envoyé sur le réseau pour trouver l'association de IP et Ethernet. **Finsi**
- **Si** événement est réception requête, envoi d'un paquet ARP Reply avec l'adresse Ethernet correspondant à l'adresse IP de la machine. **Finsi**
- **Si** événement est réception requête qui demande l'adresse Ethernet d'une adresse IP présente dans la table ARP en « Published », envoie d'un paquet ARP avec l'association demandée. **Finsi**
- **Si** événement est réception d'un paquet ARP gratuit request et que l'adresse IP source de ce paquet est inconnue. Aucune action n'est entreprise. **Finsi**
- **Si** événement est réception d'un paquet ARP gratuit request et que l'adresse IP source de ce paquet correspond à l'adresse IP de la machine courante. Un message avertissant un conflit IP est affiché sur la console. Un paquet ARP gratuit reply est envoyé pour informer l'autre machine du conflit IP. **Finsi**
- **Si** événement est réception d'un paquet ARP gratuit request et que l'adresse IP est connue, dans ce cas l'adresse Ethernet correspondante est mise à jour dans la table ARP. **Finsi**

Fin Tant que.

7) Lors d'un premier ping de A vers B, A envoi un ARP request à B pour connaître son adresse Ethernet. B répond. Cependant, B profite de cet ARP request pour enregistrer l'adresse Ethernet de A avec son adresse IP, économisant une requête ARP de sa part pour un éventuel envoi futur.

8) Un paquet ICMP est envoyé par A vers B. Aucun paquet ARP request n'est envoyé par A car cette machine possède les informations liées à B dans sa table ARP suite au ping précédent. A la réception de ce paquet ICMP par B, un ARP request est envoyé vers A. A son tour, la machine A envoie un ARP reply vers B. En effet, B ne connaît plus A car il y a eu suppression de sa table ARP. Ainsi, pour connaître l'adresse de destination du paquet ICMP reply, B doit envoyer un ARP request vers A.

9) C'est le timer de ping (1 seconde) qui re-déclenche l'émission d'un paquet ARP request. Le protocole ARP envoi juste un paquet pour demander l'adresse. Si il n'a pas de réponse, il n'y a aucune réémission.

Il est possible de tester cela avec l'envoi d'un seul paquet par le ping : `ping -c 1`. Nous constatons qu'il y a alors **un seul** paquet ARP envoyé.

RMQ : timer ARP = 1200 secondes soit 20 minutes.

10) Lors de la configuration d'une interface, un paquet ARP gratuit est envoyé sur le réseau. Un paquet ARP gratuit est un paquet envoyé en broadcast (Ethernet) avec l'adresse IP de la machine ajoutée sur le réseau. Si une machine possède déjà cette adresse IP, elle va répondre à cet ARP gratuit request par un

ARP gratuit reply pour annoncer que l'adresse IP associée à cette interface est déjà utilisée. Si il n'y a aucune réponse, pas de conflit IP.

Cet ARP gratuit permet alors de prévenir d'un conflit IP sur le réseau. Les deux machines concernées affiche un message d'alerte en console. De plus, si l'adresse IP est connue par une autre machine dans sa table ARP, l'adresse Ethernet correspondante est alors mise à jour lors de la réception de ce paquet ARP gratuit.

11) Deux paquets ARP reply sont envoyés. En effet lors d'un ping de C vers B, la machine C envoie un ARP request en broadcast (paquet No. 1 sur l'image). B reçoit ce paquet et répond avec un ARP reply (paquet No. 3 sur l'image). Cependant, la machine A possède la correspondance adresse IP et adresse Ethernet de B en published dans sa table. La clause published autorise alors A à répondre (paquet No. 2 sur l'image) au ARP request envoyé par C .Il y a donc à l'issu de ce ping deux paquets ARP reply :

1) Paquet envoyé par Machine B : source Ethernet Machine B, source ARP Machine B / paquet No.3
2) Paquet envoyé par Machine A : source Ethernet Machine A, source ARP Machine B / paquet No.2

Les adresse Ethernet source de ces deux paquets correspondent à l'adresse de la machine émettrice, cependant l'adresse source au niveau de la trame ARP est dans les deux cas l'adresse de la machine B, celle visée initialement par le ARP request (paquet No.1).

No.	Time	Source	Destination	Protocol	Length	Info
1	54.580512	Broadcom_89:e9:f5	Broadcast	ARP	60	Who has 192.168.0.2? Tell 192.168.0.3
2	54.580661	Broadcom_89:e9:ed	Broadcom_89:e9:f5	ARP	60	192.168.0.2 is at 00:10:18:89:e9:68
3	54.580662	Broadcom_89:e9:68	Broadcom_89:e9:f5	ARP	60	192.168.0.2 is at 00:10:18:89:e9:68

12) Il est possible de déclarer en published une association Ethernet/IP dans sa table ARP afin de récupérer des paquets destinés à une autre machine. En effet, lors de l'envoi d'un paquet, une machine envoie un paquet ARP request, et la machine de destination va répondre avec un ARP reply. Ensuite (il faut que se soit après le paquet ARP reply de la machine destination pour écraser l'association dans la table ARP) la machine pirate va répondre au ARP request en associant l'adresse IP de destination avec sa propre adresse Ethernet. Cela écrase donc la première association dans la table ARP de la machine source. Au prochain envoie d'un paquet, l'adresse Ethernet associée à l'adresse IP de destination sera donc l'adresse de la machine pirate.

13) Lors de la configuration d'une machine avec une adresse IP correspondant à une adresse IP déjà utilisée sur le réseau, un message d'alerte est affiché sur les deux machines correspondantes. Ce message indique que l'adresse IP est déjà utilisée ou que une machine tierce vient d'utiliser cette adresse pour configurer une de ses interfaces. Il correspond à un conflit d'IP sur le réseau :

Lors de la configuration de l'interface, un ARP gratuit request est envoyé sur le réseau. Si une machine possède déjà cette adresse IP, un message d'alerte s'affiche sur sa console et elle envoie en réponse un ARP gratuit reply à destination de la machine « voleuse » pour la prévenir du conflit. Cette dernière reçoit alors le paquet ARP reply et affiche le message d'alerte en console.

14) La machine émettrice du paquet ICMP envoie une requête ARP. Les deux machines avec la même adresse IP vont répondre en donnant leur adresse Ethernet. La première qui répond se voit retourner un

paquet ICMP (ping) car son adresse Ethernet est alors enregistrée et associée à l'adresse IP. Quand le paquet ARP reply de la deuxième machine est reçu, cela entraine l'écrasement dans la table ARP de la première association. Les paquets ICMP (ping) suivants seront tous redirigés vers la deuxième machine, celle pour qui le paquet ARP reply est arrivé en second.

Au bout de 20 minutes, le timer ARP est déclenché. Une requête ARP est lancée sur le réseau. Une fois de plus, la machine répondant en dernier sera associée à l'adresse IP pendant les prochaines 20 minutes. C'est cette dernière qui recevra les paquets ICMP reply du ping (tous sauf le premier qui sera destiné à l'autre machine, juste avant l'écrasement de son adresse MAC).

Protocole ICMP :

15) Les champs IDENTIFIER et SEQUENCE NUMBER sont identiques entre un paquet ICMP request et sa réponse le paquet ICMP reply. Dans notre cas, ils correspondent à ceux envoyés dans le fichier texte. Pour une même commande ping, le champ IDENTIFIER reste le même pour toutes les requêtes alors que le champ SEQUENCE NUMBER s'incrémente à chaque requête. Le ping est donc identifié par le champ IDENTIFIER et au sein de ce même ping, le numéro du paquet est déterminé par le champ SEQUENCE NUMBER.

Voici pour exemple le résultat d'une commande ping envoyant un total de 3 paquets :

Type de paquet	Identifier (BE)	Sequence Number (BE)
ICMP request 1	14853	0
ICMP reply 1	14853	0
ICMP request 2	14853	1
ICMP reply 2	14853	1
ICMP request 3	14853	2
ICMP reply 3	14853	2

16) Pour calculer les deux octets du checksum :

- Additionner par mot de 16 bits l'ensemble des données du paquet ICMP (sauf les deux octets du checksum)
- Réaliser le complément à 1 de ce résultat

Par exemple dans notre cas, voici les données de deux paquets ICMP :

1er paquet request : 0x 08 00 **60 8B** 12 32 85 42 00 00 00 ... (suivi d'une multitude de zéro). Les octets 60 8B sont les octets du checksum.
Il est donc nécessaire d'additionner (0x) 08 00 + 12 32 + 85 42 = 0x 9F74 (additionner les zéros suivants est bien évidemment inutile).
Puis le complément à 1 de 0x 9F74 = 0x 608B soit les octets du checksum.

2éme paquet reply : 0x 00 00 **68 8B** 12 32 85 42 ... .
Addition (0x) : 00 00 + 12 32 + 85 42 = 0x 9774.
Complément à 1 = 0x 68 8B. Correct !

Protocole DHCP :

17) Au lancement de la commande *dhclient* sur l'interface, un paquet DHCP request est envoyé par notre machine sur le réseau en broadcast Ethernet. Un paquet DHCP ACK est envoyé en réponse par le serveur DHCP. Ce paquet contient l'adresse IP du routeur par défaut et l'adresse IP que le serveur DHCP nous associe sur le réseau. Il peut également envoyer le nom du réseau (pas dans notre cas). Cette association dynamique permet de limiter les conflits IP.

il y a aussi l'adresse des serveurs DNS

Dans notre cas, aucune différence de l'interface avant et après un *dhclient* n'est constatée. Cela est probablement dû au fait que l'adressage des machines sur le réseau universitaire est statique (adresse Ethernet déjà associé à une adresse internet). Lors de la configuration de l'interface, une adresse IP est associée par défaut, identique à celle envoyé par le serveur DHCP.

18) **Fragmentation :**

Dans le cas de l'envoi d'un paquet de 4600 bytes, quatre paquets sont envoyés :

- Un premier paquet IP avec un identifiant n, une taille 1500 (data + entête IP), un flag more fragment et un fragment offset à 0.
- Un deuxième paquet IP avec un identifiant n, une taille 1500, un flag more fragment et un fragment offset à 1480.
- Un troisième paquet IP avec un identifiant n, une taille 1500, un flag more fragment et un fragment offset à 2960.
- Un paquet UDP, avec en IP un identifiant n, une taille 188 (data + entête IP + en tête UDP) et un fragment offset à 4440. Dans la trame UDP, il y a la taille total du paquet non fragmenté de 4608 (data + en tête UDP).

L'entête UDP se situe sur le premier paquet. IP rassemble alors du premier paquet au dernier paquet. Après réassemblage, il ne reste qu'un paquet UDP de 4608 bytes (4600 de data et 8 d'entête UDP).

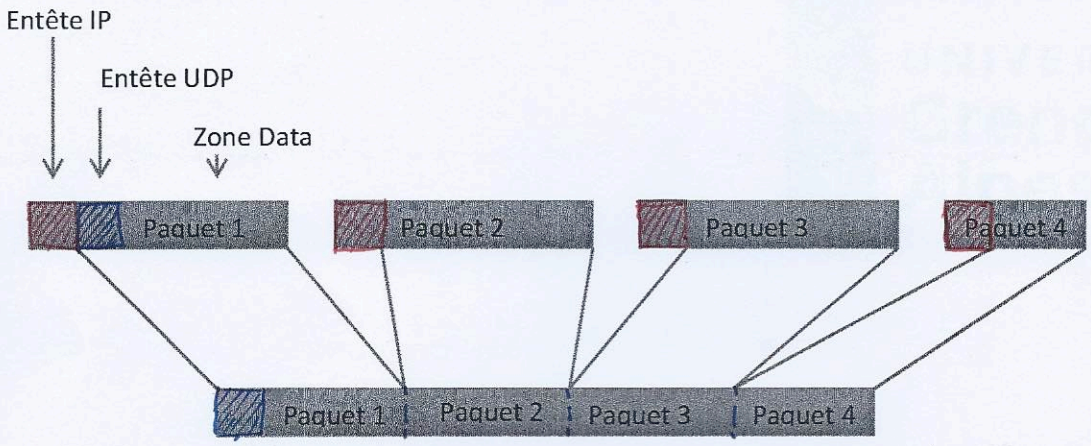
La capture de paquet sur wireshark est la suivante :

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.2	192.168.0.3	IPv4	1514	Fragmented IP protocol
2	0.001241	192.168.0.2	192.168.0.3	IPv4	1514	Fragmented IP protocol
3	0.002490	192.168.0.2	192.168.0.3	IPv4	1514	Fragmented IP protocol
4	0.002646	192.168.0.2	192.168.0.3	UDP	202	25724 → 13000 Len=4600

Cette capture pourrait laisser croire que l'entête UDP se trouve sur le dernier paquet (No.4). Cependant, en analysant le contenu de la zone data de chaque paquet, il s'avère que l'entête UDP se trouve effectivement dans le premier paquet ainsi que les premiers bytes de donnée de l'application. A la réception du dernier paquet (No.4, sans le flag : MORE FRAGMENT indiquant que c'est le dernier paquet), IP réassemble les paquets. Wireshark indique alors un paquet UDP, car il indique le type du paquet réassemblé.

ok

Réception des paquets (les paquets sont représentés ici à partir de la couche IP) + réassemblage :



Au réassemblage, les entêtes IP intermédiaires sont supprimées, et le paquet UDP est réassemblé comme représenté par la figure ci-dessus.