**M2-CCI Enseignement de Réseaux – Compte-Rendu du TP n.2**

*Groupe BAJA Sara – BOLLOLI Marco*

**Introduction :**

Ce compte-rendu a pour objet les manipulations effectuées dans le TP n.2, et les observations et les calculs qui y étaient associés. De suite, les étapes et les questions seront présentées avec la numérotation adoptée dans l’énoncé du TP pour des raisons de clarté.

**Etude des protocoles de niveau 3 (2.1) *:***

De façon très similaire à celle du TP1, le réseau est mis en place à partir de 4 ordinateurs et un Hub tel que décrit dans la figure suivante.

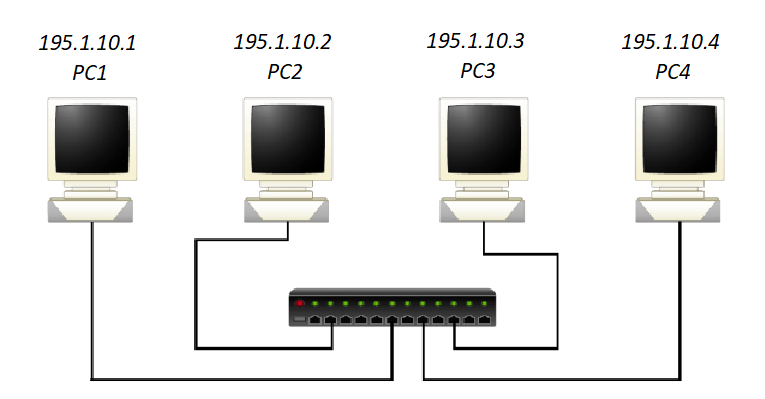


Figure 1

Dans chaque ordinateur, nous n’utilisons pas l’interface réseau de la carte mère (*em0*) mais plutôt l’interface *bge0* qui est installée en plus et qui permet d’observer des collisions entre paquets.

L’interface n’étant pas associé à une adresse IP v4, nous avons procédé à le faire pour chaque ordinateur à l’aide de la commande **ifconfig <nom interface> <adresse>**. De suite le paramètres choisis :

***Machine 1****: 195.1.10.1*

***Machine 2****:195.1.10.2*

***Machine 3****:195.1.10.3*

***Machine 4****:195.1.10.4*

Nous avons bien vérifié, à l’aide de la même commande ifconfig, que bge0 était correctement configurée sur chaque ordinateur (Cf. Figure 2).

Les informations qui nous intéressent les plus ici sont celles données par le **status** et le **media**.  Le fait que **status** est en mode active et qu’on a 10base/UTP <half-duplex> au niveau de **média** indique que l’interface est en marche.

Sauf où autrement spécifié, le *PC3* est la machine utilisée pour suivre les manipulations et enregistrer le passage des paquet parmi les autres 3 machines du réseau.

**2.1.1 Le protocole ARP**

Après avoir vidé les tables ARP, nous lançons un Ping de *PC1* à *PC2* et *PC2* à *PC4*.

Nous ouvrons WireShark sur *PC3* pour enregistrer les échanges.

**Question 1 : Qu'est ce qui permet d'identifier les paquets comme étant de type ARP?**

Dans la Figure 1, nous voyons la capture correspondante à un Ping entre *PC2* et *PC4*. Nous voyons que d’abord *PC2* envoie un paquet ARP à tout le réseau à travers l’adresse *broadcast* (ff : ff : ff : ff : ff : ff), pour trouver *PC4*. Le fait qu’il agit bien d’un ARP est indiqué par le EtherType x0806.

**Question 2 : Où se situe le paquet ARP dans la trame Ethernet ?**

Le paquet se situe après le EtherType, dans la zone Data. Dans la Figure 1 en bas, nous pouvons voire le paquet ARP (sur fond blanc) et le padding (sur fond bleu) juste après.

**Question 3 : De la même manière, faites le schéma correspondant au paquet ARP. Indiquez le rôle de chacun des champs du paquet ARP.**

Le schéma suivant correspondant au paquet ARP. Le nombre d’octets des champs sont indiqués entre parenthèses. AM: adresse machine, IP: adresse internet.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Type  matériel  (2) | Type de protocole  (2) | longueur adresse  matériel  (1) | longueur  adresse  de protocole  (1) | type d’opération  (2) | Source  AM  (6) | Source IP  (4) | Destination  AM  (6) | Destination  IP  (4) |

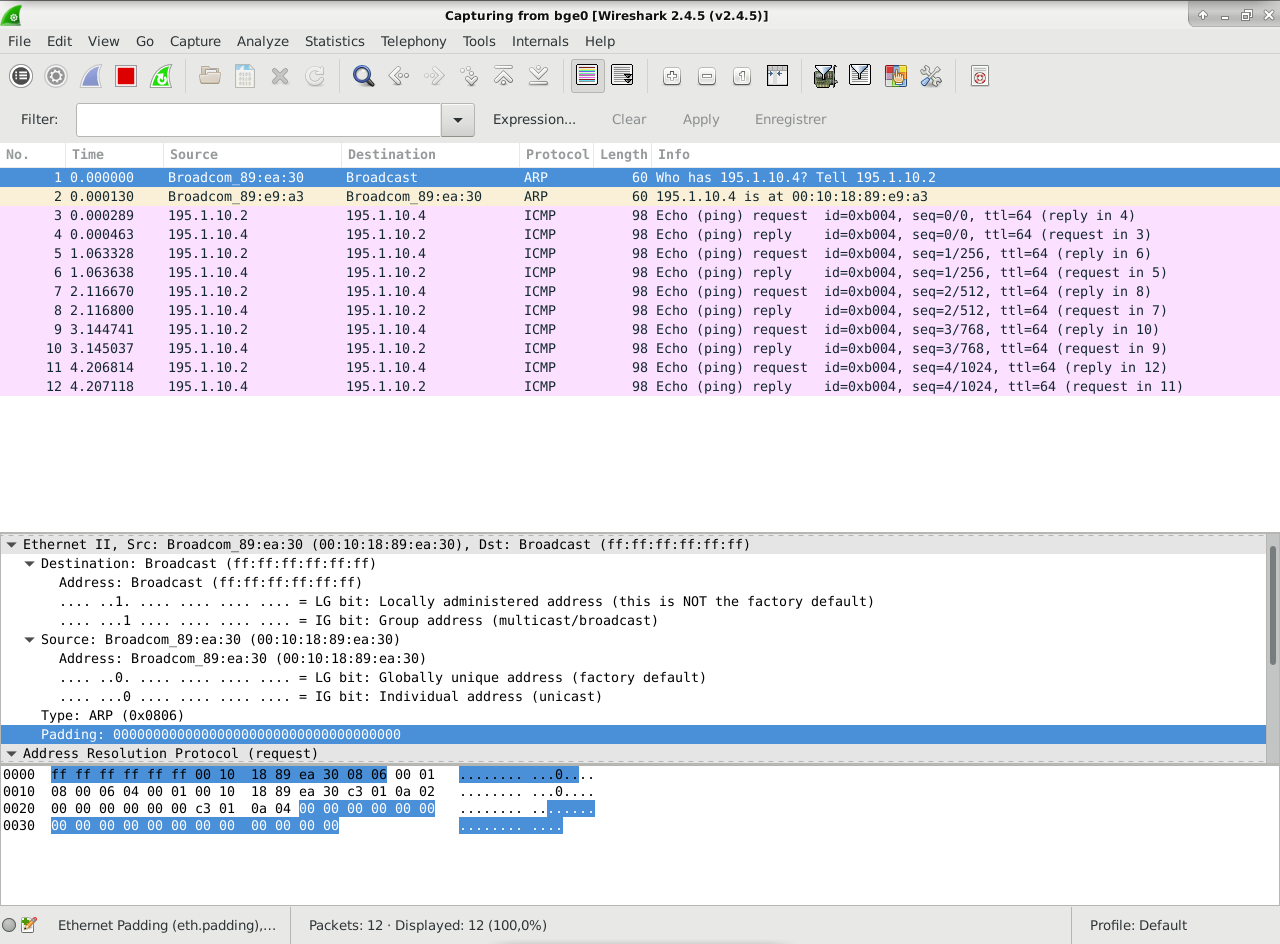


Figure 1 (questions 2 et 4)

**Question 4 : Comment le niveau Ethernet connait-il la taille des trames qu'il recoit ? Comment determine-t-il la fin de la trame ? Retrouvez les octets de bourrage d'Ethernet dans la trame.**

Le niveau Ethernet ne connait pas la taille des paquets envoyés. A ce niveau, le paquet est identifié par le début du signal qui lui est associé. Le début du paquet est indiqué par un marqueur particulier et la fin par un silence.

Lorsque la taille du paquet est inférieure à 64 octets, on ajoute des octets de bourrage qui sont constitués par des zéros. Sur la figure 1, on voit bien les octets de bourrage (en bas, sur fond bleu) à la fin du paquet.

**Question 5 : Qu'est ce qui est transmis a la couche ARP par la couche Ethernet (à la réception d'un paquet) ? Ethernet connait-il l'existence d'un bourrage à la réception d'un paquet ?**

La couche Ethernet transmet à la couche ARP tout le paquet ARP + le bourrage, donc une réponse qui contient l’adresse de la machine demandée. À la réception d’un paquet, le protocole Ethernet ne connait pas des octets de bourrage. Ils sont connus par les protocoles des couches supérieures. Après un ping de A vers B. Nous observons des paquets de type ARP. A envoie un paquet de type ARP-request en demandant l’adresse de la machine B. B envoie un paquet de type ARP-reply pour donner son adresse machine à A.

**Question 6 : compléter de façon informelle le corps de l'algorithme suivant**

Tant que(1) Faire

Attendre un évènement x

Si x = Envoi d’un paquet vers une station Alors

Si son adresse est connue Alors

Envoyer le message à cette adresse

Sinon

Envoyer un paquet ARP Broadcast pour récupérer son adresse

Fin si

Fin si

Si x = Arrivée d’une requête ARP Alors

Si l’émetteur est dans la table ARP Alors

Remettre à 0 le compteur de son entrée

Sinon

L’ajouter à la table

Fin si

Si la requête m’est adressée Alors

Envoyer une réponse ARP

Sinon Si je connais le destinataire et que son entrée est published Alors

Envoyer une réponse ARP avec son adresse MAC

Fin si

Fin si

Si x = Arrivée d’une réponse ARP Alors

Ajouter l’entrée dans la table ARP

Fin si

Si x = Le timer d’une entrée arrive à expiration Alors

Supprimer l’entrée de la table ARP

Fin si

Fin Tant que

**Question 7 : Observez les tables ARP de A et B. Pourquoi B apparait-il dans la table de A? Comment B a t-il pu apprendre l'adresse de A alors qu'il n'a pas envoyé de ARP-request à A?**

Lors de l’envoi d’un ARP- request, A insère dans le message son adresse (l’adresse source) : cela permet à la machine réceptrice de connaitre la machine à laquelle envoyer le paquet ARP-reply.

**Question 8 : Effacez la table ARP de B. Recommencez le ping de A vers B. Notez les paquets engendrés. Expliquez.**

A maintenant connait l’adresse de B, donc il lui envoie directement un paquet ICMP. Par contre, B ne connait pas de qui ce paquet arrive, donc il est contraint d’envoyer un paquet ARP en broadcast pour l’adresse ethernet de la machine émettrice. (Cf. Figure 2)

Après réception de la réponse (paquet ARP-reply), B peut enfin renvoyer son paquet ICMP.

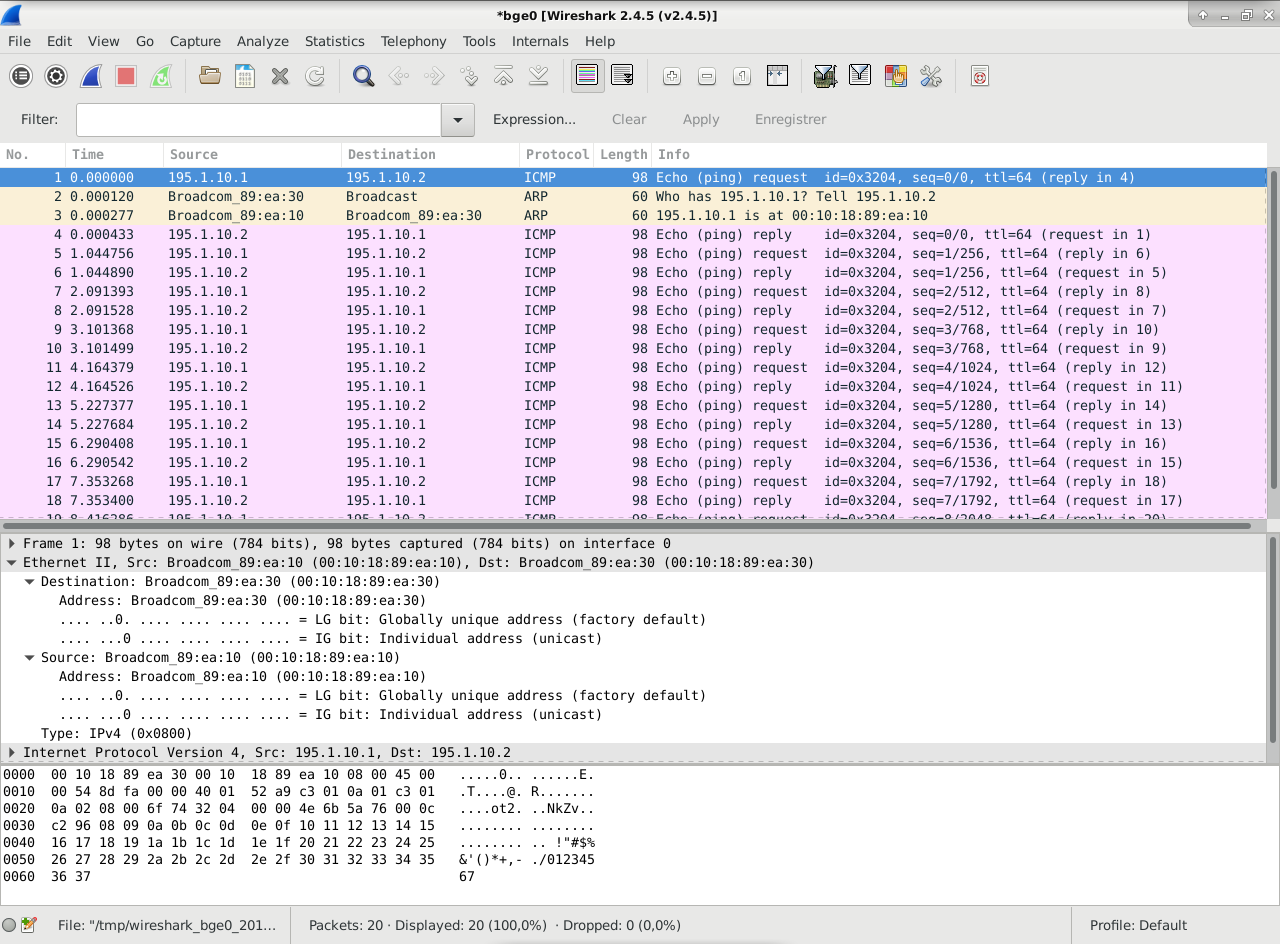


Figure 2 (question 8)

**Question 9 : Est ce que c'est le timer de ARP ou de Ping qui déclenche les réémissions des paquets ARP-Request ? Quelle est la durée de ce timer de réémission ?**

Comme A ne connait pas B, mais B est debranché (donc ne peut pas répondre) le ping de A vers B après avoir débranché la machine B déclenche des émissions des paquets ARP-request. Ces émissions sont déclenchées par le timer du ping non par celui de ARP. Le timer de ARP est de 1200 s (20 minutes), mais comme nous observons un timer plus rapide, celui-ci ne peut correspondre qu’à celui de ping (environ 1s).

**Question 10 : Observez le réseau au moment où vous configurer une interface (if-config). À quoi peut servir le ARP gratuit ?**

Les Paquets ARP servent à signaler l’adresse IP d’un nouvel ordinateur sur le réseau. L’ARP gratuit request est envoyé à toutes les machines du réseau pour savoir s’il existe une machine qui a la même adresse internet. Si l’adresse n’existe pas encore, on n’a pas d’ARP gratuit reply.

**Question 11 : Pourquoi deux paquets de type ARP reply sont-ils observés ? Expliquez ce qu'il s'est passé (demandez une analyse détaillée de ces paquets ; étudiez en particulier les adresses Ethernet et Internet).**

Les adresses IP published sont ″publiques″, donc si une machine connait une adresse published elle répondra à une ARP-request, même si ce n’est pas son propre adresse Ethernet. Cela car la correspondance à l’adresse Ethernet est gérée à bas niveau, alors que la correspondance avec l’adresse IP est gérée au niveau ARP (donc la machine repond ″ne sachant pas″ que ce n’est pas son ″vrai″ IP). Suite à plusieurs ARP-reply avec la meme IP, ce sera la dernière réponse emise qui sera rétenue dans la table ARP du récepteur.

Donc si PC1 a mémorisé les informations (Ethernet et IP) de PC2, suite à l’envoi d’une ARP-request de la part de PC4, pour demander qui a l’IP 190.1.10.2 (IP de PC2), il y aura 2 paquets ARP-reply qui seront envoyés, de la part de PC1 et PC2. (Cf Figure 11) Les entêtes Ethernet seront différents au niveau des octets de l’adresse source, par contre le paquet ARP contiendra les mêmes informations, donc PC4, dans ce cas, mémorisera la bonne information dans sa table ARP.

**Question 12 : Dans quels cas la déclaration d'une adresse en Publish d'une autre machine peut-elle être intéressante ? (avec une autre adresse Ethernet par exemple).**

Pour des raisons de sécurité ou administration réseau, une machine pourrait prendre l’adresse IP d’une autre et donc intercepter ou bloquer l’envoi des données vers celle-ci.

**Question 13 : Au moment où vous configurez la deuxième machine, vous devriez avoir un message du système à l'écran (si vous sortez de l'environnement "xinit" ) sur la première. A quoi sert-il ? Qu'est ce qui l'a déclenché?**

Nous avons bien visualisé le message. Le message indique que l’adresse IP est utilisée par une autre machine et il est déclenché après l’envoi d’un paquet ARP gratuit après la nouvelle configuration de la machine. Ceci sert à eviter que plusieurs machines aient le même IP sur un même réseau.

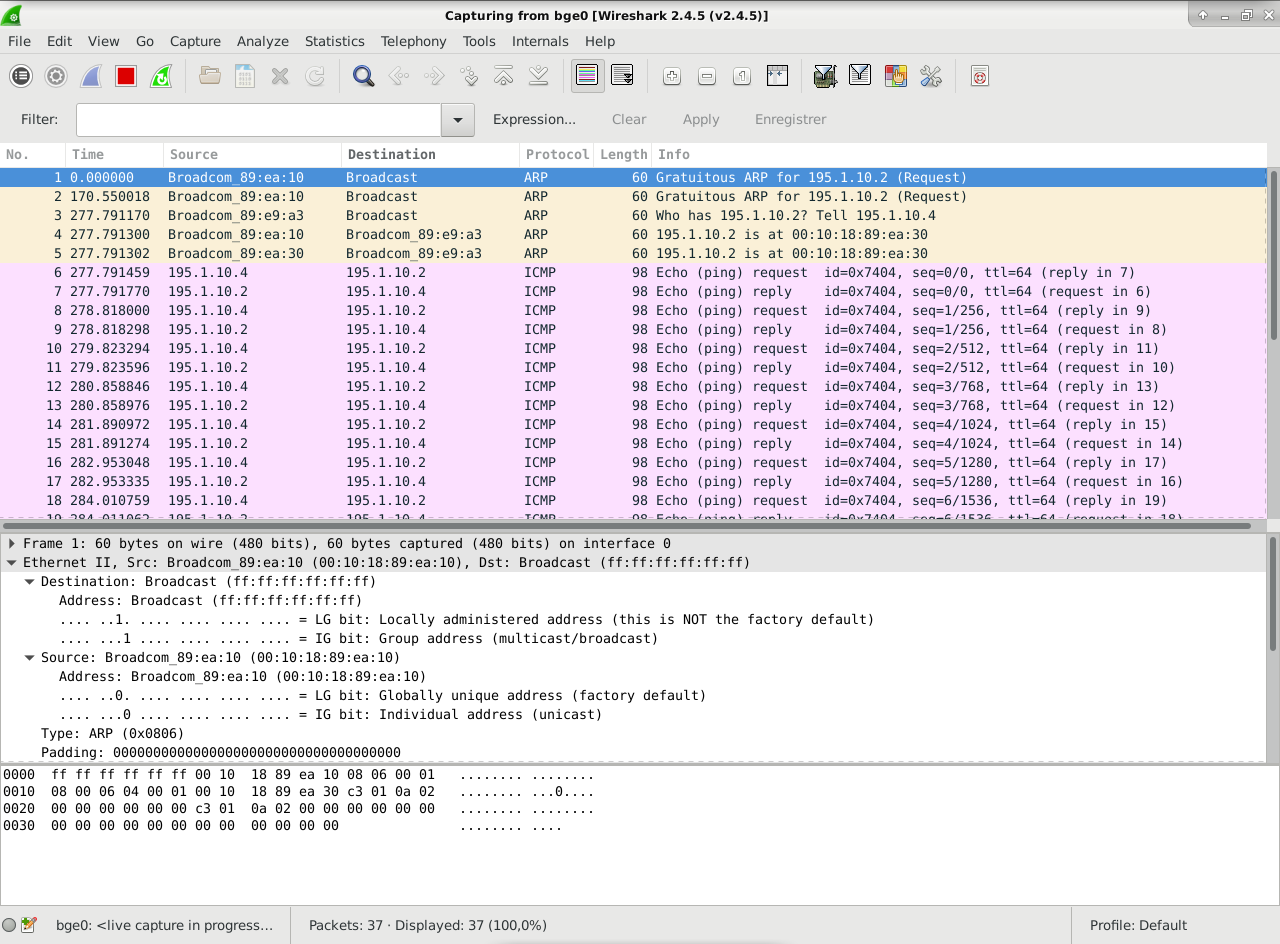
****

Figure 3 (question 11)

**Question 14 : Expliquez ce qu'il se passe quand on fait un ping vers l'adresse IP commune ? Conclusions ?**

Dans ce cas-ci, PC1 a pris l’IP de PC2 et il l’a associé à sa propre adresse Ethernet. Lors d’un ping de PC4 vers l’IP de PC2, les deux machines répondront, les ping suivants seront fait seulement entre PC4 et une des 2 machines.

On observe qu’il peut se produire (pas à tous les coups) ce phénomène : PC4 continue à envoyer vers une machine, mais c’est l’autre qui lui répond (car elle a maintenant le meme IP), pas nécessairement B. Donc A a effectivement usurpé l’adresse de B.

**2.1.2 Le protocole ICMP**

**Question 15 : Que remarquez-vous pour le champ Identifer et le champ Sequence Number ? Observez ces deux champs dans les paquets générés par un ping ? A quoi servent-ils ?**

Dans notre fichier pour l’envoi ICMP, nous écrivons (en hexadécimale) :

Type: Echo request = 08

Code-type ( pour une demande ou une réquete): 00

checksum : 00 00 (pour faciliter le calcul au départ)

identifier ID: 01 01

sequence Number: 02 02

Donc la séquence se présente comme 08 00 00 00 01 01 02 02

Après envoi du fichier de *PC1* à *PC4* à l’aide de la commande **send\_icmp,** le fichier reçu par *PC4* présente une structure légèrement différente, car le checksum a été modifié par le protocole CRC pour la détection d’erreur (Cf. Figure 4).

Concernant les champs Identifier et Sequence Number, le premier correspond au numéro de série du ping qui servira à identifier si c’est un request ou reply, tandis que le deuxième correspond au numéro du ping dans la série.

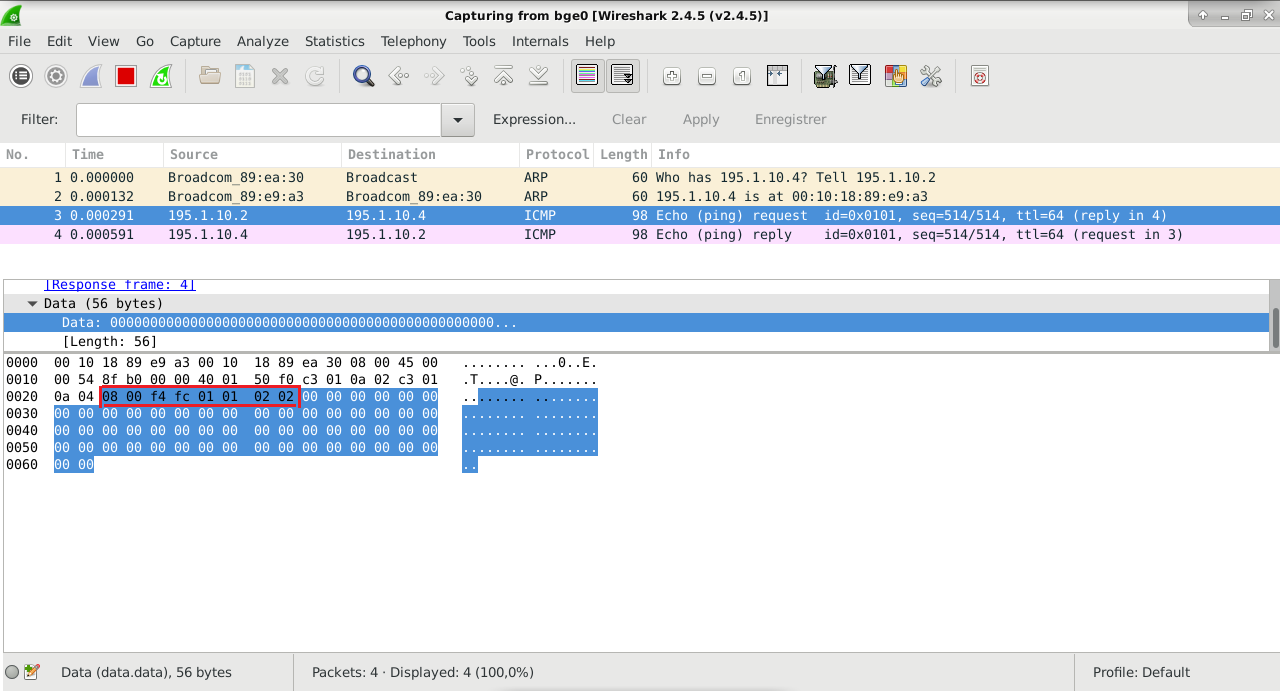


Figure 4 (questions 15 et 16)

**Question 16 : Donnez l'algorithme de calcul des deux octets de Checksum. Vérifiez sur un de vos paquets ICMP capturés ce calcul. Quels types d'erreurs ce type de vérification peut-elle détecter et ne pas détecter ?**

Un checksum est un algorithme qui permet de donner dans le message une indication d’erreurs de transmission de message. On ajoute les éléments de la trame, deux octets à deux octets, et ensuite nous calculons le complément à 1

Dans notre exemple : 08 00 00 00 01 01 02 02.

08 00

01 01

02 02

--------

0B 03 🡺 f4 fc checksum qui effectivement apparait dans le paquet ICMP.

L’avantage de cette méthode, surtout pour un checksum sur des lignes longues (16 ou 32 bits) est la possibilité d’identifier des erreurs en rafales. La limite est que deux erreurs isolés peuvent se compenser, ne pas modifier le checksum, donc passer la detection.

**2.2 Etude du protocole DHCP**

**Question 17 : Observez et commentez les paquets capturés alors. Expliquez en particulier les informations contenues dans le paquet DHCP ACK. Aidez vous du man de dhclient et du cours pour expliquer le contenu de ces paquets. Expliquez la nouvelle configuration de l'interface em0.**

On connecte PC2 à la prise murale via l’interface em0. Ensuite on utilise la commande **dhclient** **em0**. La machine envoie d’abord en broadcast une trame DHCPDISCOVER pour trouver les serveurs disponibles. Un ou plusieurs serveurs répondent avec une trame DHCPOFFER, un proposant une adresse libre dan leur réseau. Le client renvoie un paquet DHCPREQUEST pour confirmer le serveur duquel il accepte l’offre, voire pour demander d’autres paramètres, et le serveur lui envoie un DHCP ACK qui confirme l’adresse IP exacte.

Dans notre cas, l’adresse IP attribué et le 152.77.84.22

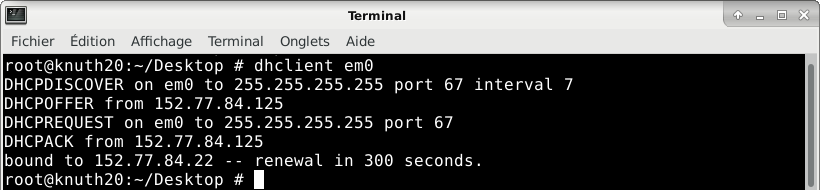
****

Figure 5 (question 17)

**2.3 Étude de la fragmentation des paquets IP**

**Question 18 : Analysez les paquets engendrés sur le réseau. En particulier, étudiez les entetes IP et expliquez les rôles des champs IDENTIFICATION, TOTAL LENGTH, FRAGMENT OFFSET, et du flag MF ("More Fragment"). Répéter si nécessaire la manipulation avec d'autres tailles de paquets. Pourquoi l'entête UDP n'apparait que dans un seul paquet ? Dans quel ordre sont envoyés les fragments ?**

**Retrouvez dans l'hexadecimal des paquets l'entête UDP. L'entête UDP est elle dans le premier ou le dernier fragment ? Retrouvez précisément dans ces paquets les 4600 octets de données de l'application. Faites une figure pour expliquer le découpage effectué par IP.**

Nous procédons à un envoi à travers le protocole UDP d’un paquet de 4600 octets, ce qui force le protocole IP à le fragmenter en 4 paquets.

L’entête UDP apparait dans un seul fragment car le découpage est effectué après avoir mis l’entête. Elle est envoyé donc une seul fois, avec le premier fragment (les fragments sont envoyés en ordre de réassemblage, sauf si l’utilisateur le réordonne).

Concernant les différentes champs :

Identification : identifie le paquet par tous les paquets, pour le réassembler à la fin ;

More Fragment : MF=1 veut dire qu’il y a encore des paquets à venir, si MF=0 alors c’est le dernier paquet ;

Total lenght : longueur du fragment ;

Fragment Offset : le numéro du premier octet du fragment par rapport à la taille du paquet original (ex. dans la Figure 6 nous pouvons voire que le paquet original à été fragment aux octets 1480 et 2960)

**SI nous vouslong faire un pétit schéma de l’envoi**

|  |  |  |
| --- | --- | --- |
| entete  IP | Entete  UDP | Data |

|  |  |
| --- | --- |
| Entete  UDP | Data |

|  |  |
| --- | --- |
| Entete  UDP | Data |

|  |  |
| --- | --- |
| Entete  UDP | Data |

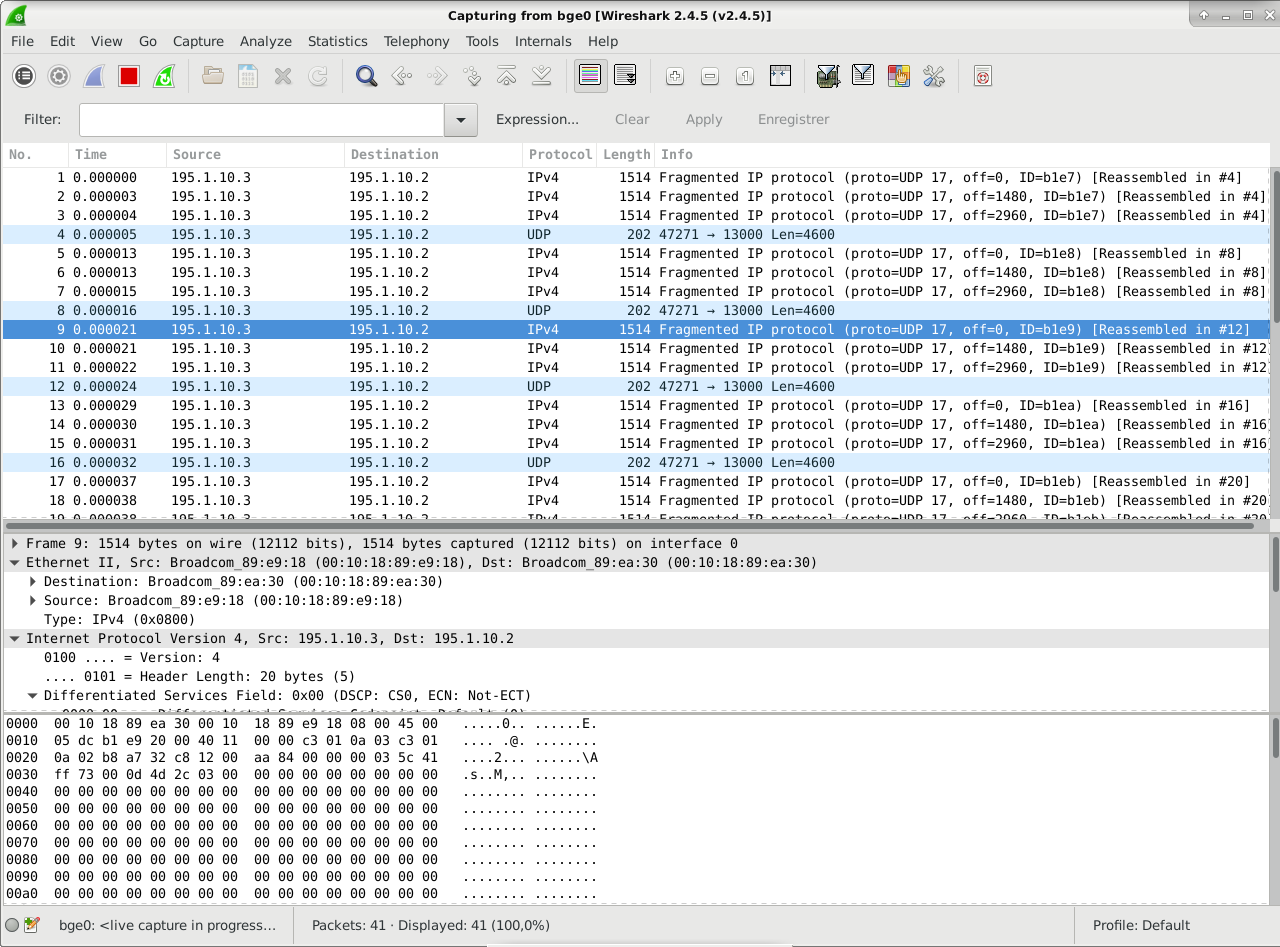
****

Figure 6 (question 18)

**Conclusion :**

Ce TP nous permis d’avoir des notions de base sur le fonctionnement d’un réseau Ethernet. Nous avons pu constater quelles sont les différentes composantes du temps de transmission des données d’une machine à une autre, et nous avons-nous initier à la gestion d’un réseau avec plusieurs machines connectées ensemble.