

Travaux Dirigés
Réseaux (modèle TCP/IP)

Encapsulation et analyse des échanges

Cheminement des trames et routage

La couche Réseau

La couche Transport

TD1 Etude du modèle TCP/IP

L'objectif est d'une part d'étudier l'accès à un réseau ethernet et le phénomène d'**encapsulation** sur la pile de protocoles TCP/IP et d'autre part **d'analyser les échanges** ayant lieu entre deux machines.

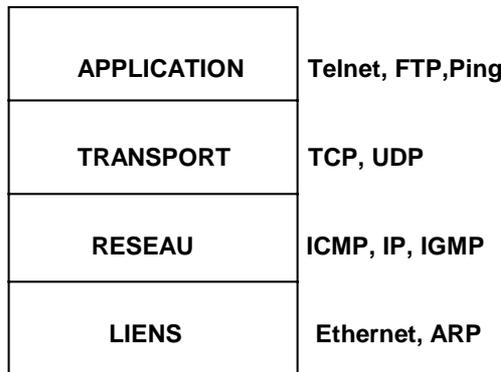


figure (1) : Organisation en couches du protocole TCP/IP

A- On suppose trois stations A, B et C désirant émettre en même temps sur leur réseau ethernet. Sachant que le protocole utilisé pour la couche MAC est le CSMA/CD, expliquer la résolution du conflit d'accès au médium à travers un scénario (ex: A puis C puis B).

B- A partir des trames représentées dans les tableaux ci-joints, étudiez les échanges ayant lieu entre les deux machines impliquées :

- 1) Indiquer de quelle application il s'agit et préciser les protocoles utilisés aux différents niveaux du modèle.
- 2) Quelles sont les opérations principales réalisées (couches transport et application) ?
- 3) En indiquant les numéros de séquences et d'acquittement, tracer les différentes opérations.
- 4) Sur quel port les données sont-elles dirigées ? Est-ce le même que celui utilisé dans un premier temps par l'application ?
- 5) Est-ce que le nom de l'utilisateur et son mot de passe sont cryptés ?

<i>trame 1</i>	<i>trame 2</i>
ETHER: ----- Ether Header ----- ETHER: Packet 1 arrived at 17:33:20.05 ETHER: Packet size = 58 bytes ETHER: Destination = 0:0:c:3a:4c:ce, Cisco ETHER: Source = 8:0:20:74:8c:a0, Sun ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- IP: Version = 4 IP: Header length = 20 bytes IP: Type of service = 0x00 IP: xxx. = 0 (precedence) IP: ...0 = normal delay IP: 0... = normal throughput IP: 0.. = normal reliability IP: Total length = 44 bytes IP: Identification = 59245 IP: Flags = 0x4 IP: .1.. = do not fragment IP: ..0. = last fragment IP: Fragment offset = 0 bytes IP: Time to live = 255 seconds/hops IP: Protocol = 6 (TCP) IP: Header checksum = bef6 IP: Source address = 194.199.229.99, gtr-server1 IP: Destination address=193.49.108.10, simone.lirmm.fr IP: No options IP:	ETHER: ----- Ether Header ----- ETHER: Packet 2 arrived at 17:33:20.08 ETHER: Packet size = 60 bytes ETHER: Destination = 8:0:20:74:8c:a0, Sun ETHER: Source = 0:0:c:3a:4c:ce, Cisco ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- IP: Version = 4 IP: Header length = 20 bytes IP: Type of service = 0x00 IP: xxx. = 0 (precedence) IP: ...0 = normal delay IP: 0... = normal throughput IP: 0.. = normal reliability IP: Total length = 40 bytes IP: Identification = 54653 IP: Flags = 0x0 IP: .0.. = may fragment IP: ..0. = last fragment IP: Fragment offset = 0 bytes IP: Time to live = 52 seconds/hops IP: Protocol = 6 (TCP) IP: Header checksum = dbeb IP: Source address = 193.49.108.10, simone.lirmm.fr IP: Destination address= 194.199.229.99, gtr-server1 IP: No options IP:

<i>Suite trame 1</i>	<i>Suite trame 2</i>
<pre>TCP: ----- TCP Header ----- TCP: Source port = 33185 TCP: Destination port = 21 (FTP) TCP: Sequence number = 192020081 TCP: Acknowledgement number = 0 TCP: Data offset = 24 bytes TCP: Flags = 0x02 TCP: ..0. = No urgent pointer TCP: ...0 = No acknowledgement TCP: 0... = No push TCP: 0.. = No reset TCP: 1. = Syn TCP: 0 = No Fin TCP: Window = 8760 TCP: Checksum = 0x14ee TCP: Urgent pointer = 0 TCP: Options: (4 bytes) TCP: - Maximum segment size = 1460 bytes TCP: FTP: ----- FTP: ----- FTP: "" FTP:</pre>	<pre>TCP: ----- TCP Header ----- TCP: Source port = 21 TCP: Destination port = 33185 TCP: Sequence number = 1347648000 TCP: Acknowledgement number = 192020082 TCP: Data offset = 20 bytes TCP: Flags = 0x12 TCP: ..0. = No urgent pointer TCP: ...1 = Acknowledgement TCP: 0... = No push TCP: 0.. = No reset TCP: 1. = Syn TCP: 0 = No Fin TCP: Window = 4096 TCP: Checksum = 0x747d TCP: Urgent pointer = 0 TCP: No options FTP: ----- FTP: ----- FTP: "" FTP:</pre>

<i>trame 3</i>	<i>trame 4</i>
<pre>ETHER: ----- Ether Header ----- ETHER: Packet 3 arrived at 17:33:20.08 ETHER: Packet size = 54 bytes ETHER: Destination = 0:0:c:3a:4c:ce, Cisco ETHER: Source = 8:0:20:74:8c:a0, Sun ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- IP: Version = 4 IP: Header length = 20 bytes IP: Type of service = 0x00 IP: xxx. = 0 (precedence) IP: ...0 = normal delay IP: 0... = normal throughput IP: 0.. = normal reliability IP: Total length = 40 bytes IP: Identification = 59246 IP: Flags = 0x4 IP: .1.. = do not fragment IP: ..0. = last fragment IP: Fragment offset = 0 bytes IP: Time to live = 255 seconds/hops IP: Protocol = 6 (TCP) IP: Header checksum = bef9 IP: Source address = 194.199.229.99,gtr-server1 IP: Destination address=193.49.108.10,simone.lirmm.fr IP: No options IP: TCP: ----- TCP Header ----- TCP: Source port = 33185 TCP: Destination port = 21 (FTP) TCP: Sequence number = 192020082 TCP: Acknowledgement number = 1347648001 TCP: Data offset = 20 bytes TCP: Flags = 0x10 TCP: ..0. = No urgent pointer TCP: ...1 = Acknowledgement TCP: 0... = No push TCP: 0.. = No reset TCP: 0. = No Syn TCP: 0 = No Fin TCP: Window = 9112 TCP: Checksum = 0x60e6 TCP: Urgent pointer = 0 TCP: No options TCP: FTP: ----- FTP: ----- FTP: "" FTP:</pre>	<pre>ETHER: ----- Ether Header ----- ETHER: Packet 4 arrived at 17:33:20.17 ETHER: Packet size = 105 bytes ETHER: Destination = 8:0:20:74:8c:a0, Sun ETHER: Source = 0:0:c:3a:4c:ce, Cisco ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- IP: Version = 4 IP: Header length = 20 bytes IP: Type of service = 0x00 IP: xxx. = 0 (precedence) IP: ...0 = normal delay IP: 0... = normal throughput IP: 0.. = normal reliability IP: Total length = 91 bytes IP: Identification = 54655 IP: Flags = 0x0 IP: .0.. = may fragment IP: ..0. = last fragment IP: Fragment offset = 0 bytes IP: Time to live = 52 seconds/hops IP: Protocol = 6 (TCP) IP: Header checksum = dbb6 IP: Source address = 193.49.108.10, simone.lirmm.fr IP: Destination address = 194.199.229.99,gtr-server1 IP: No options IP: TCP: ----- TCP Header ----- TCP: Source port = 21 TCP: Destination port = 33185 TCP: Sequence number = 1347648001 TCP: Acknowledgement number = 192020082 TCP: Data offset = 20 bytes TCP: Flags = 0x18 TCP: ..0. = No urgent pointer TCP: ...1 = Acknowledgement TCP: 1... = Push TCP: 0.. = No reset TCP: 0. = No Syn TCP: 0 = No Fin TCP: Window = 4096 TCP: Checksum = 0x7e5e TCP: Urgent pointer = 0 TCP: No options TCP: FTP: ----- FTP: ----- FTP: "220 simone.lirmm.fr FTP server (SunOS 4.1) ready.\r\n" FTP:</pre>

<i>Trame 10</i>	<i>trame 11</i>
<pre>ETHER: ----- Ether Header ----- ... ETHER: Destination = 0:0:c:3a:4c:ce, Cisco ETHER: Source = 8:0:20:74:8c:a0, Sun ETHER: Ethertype = 0800 (IP) ETHER:</pre>	<pre>ETHER: ----- Ether Header ----- ... ETHER: Destination = 8:0:20:74:8c:a0, Sun ETHER: Source = 0:0:c:3a:4c:ce, Cisco ETHER: Ethertype = 0800 (IP) ETHER:</pre>

<p style="text-align: center;">Suite trame 10</p> <pre> IP: ----- IP Header ----- ... IP: Protocol = 6 (TCP) IP: Header checksum = bee9 IP: Source address = 194.199.229.99, gtr-server1 IP: Destination address=193.49.108.10,simone.lirmm.fr IP: No options IP: TCP: ----- TCP Header ----- TCP: Source port = 33185 TCP: Destination port = 21 (FTP) TCP: Sequence number = 192020082 TCP: Acknowledgement number = 1347648052 ... FTP: ----- FTP: ----- FTP: "USER andreu\r\n" FTP: </pre>	<p style="text-align: center;">Suite trame 11</p> <pre> IP: ----- IP Header ----- ... IP: Protocol = 6 (TCP) IP: Header checksum = dbc3 IP: Source address = 193.49.108.10, simone.lirmm.fr IP: Destination address=194.199.229.99, gtr-server1 IP: No options IP: TCP: ----- TCP Header ----- TCP: Source port = 21 TCP: Destination port = 33185 TCP: Sequence number = 1347648052 TCP: Acknowledgement number = 192020096 ... FTP: ----- FTP: ----- FTP: "331 Password required for andreu.\r\n" FTP: </pre>
---	--

<p style="text-align: center;">trame 12</p> <pre> ETHER: ----- Ether Header ----- ... ETHER: Destination = 0:0:c:3a:4c:ce, Cisco ETHER: Source = 8:0:20:74:8c:a0, Sun ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- ... IP: Source address = 194.199.229.99, gtr-server1 IP: Destination address= 193.49.108.10,simone.lirmm.fr IP: No options IP: TCP: ----- TCP Header ----- TCP: Source port = 33185 TCP: Destination port = 21 (FTP) TCP: Sequence number = 192020096 TCP: Acknowledgement number = 1347648088 ... FTP: ----- FTP: ----- FTP: "PASS yapuka\r\n" FTP: </pre>	<p style="text-align: center;">trame 13</p> <pre> ETHER: ----- Ether Header ----- ... ETHER: Destination = 8:0:20:74:8c:a0, Sun ETHER: Source = 0:0:c:3a:4c:ce, Cisco ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- ... IP: Source address = 193.49.108.10, simone.lirmm.fr IP: Destination address = 194.199.229.99, gtr-server1 IP: No options IP: TCP: ----- TCP Header ----- TCP: Source port = 21 TCP: Destination port = 33185 TCP: Sequence number = 1347648088 TCP: Acknowledgement number = 192020109 ... FTP: ----- FTP: ----- FTP: "230 User andreu logged in.\r\n" FTP: </pre>
---	--

<p style="text-align: center;">trame 25</p> <pre> ETHER: ----- Ether Header ----- ... ETHER: Destination = 8:0:20:74:8c:a0, Sun ETHER: Source = 0:0:c:3a:4c:ce, Cisco ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- ... IP: Source address = 193.49.108.10, simone.lirmm.fr IP: Destination address = 194.199.229.99, gtr-server1 IP: No options IP: TCP: ----- TCP Header ----- TCP: Source port = 20 TCP: Destination port = 33186 TCP: Sequence number = 1350400000 TCP: Acknowledgement number = 0 ... FTP-DATA: ----- FTP-DATA: ----- FTP-DATA: "" FTP-DATA: </pre>	<p style="text-align: center;">trame 26</p> <pre> ETHER: ----- Ether Header ----- ... ETHER: Destination = 0:0:c:3a:4c:ce, Cisco ETHER: Source = 8:0:20:74:8c:a0, Sun ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- ... IP: Source address = 194.199.229.99, gtr-server1 IP: Destination address=193.49.108.10,simone.lirmm.fr IP: No options IP: TCP: ----- TCP Header ----- TCP: Source port = 33186 TCP: Destination port = 20 (FTP-DATA) TCP: Sequence number = 194836081 TCP: Acknowledgement number = 1350400001 ... FTP-DATA: ----- FTP-DATA: ----- FTP-DATA: "" FTP-DATA: </pre>
--	--

<p style="text-align: center;">trame 39</p> <pre> ETHER: ----- Ether Header ----- ... ETHER: Destination = 0:0:c:3a:4c:ce, Cisco ETHER: Source = 8:0:20:74:8c:a0, Sun ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- ... IP: Source address = 194.199.229.99, gtr-server1 IP: Destination address= 193.49.108.10,simone.lirmm.fr IP: No options IP: TCP: ----- TCP Header ----- TCP: Source port = 33185 TCP: Destination port = 21 (FTP) </pre>	<p style="text-align: center;">trame 40</p> <pre> ETHER: ----- Ether Header ----- ... ETHER: Destination = 8:0:20:74:8c:a0, Sun ETHER: Source = 0:0:c:3a:4c:ce, Cisco ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- ... IP: Source address = 193.49.108.10, simone.lirmm.fr IP: Destination address = 194.199.229.99,gtr-server1 IP: No options IP: TCP: ----- TCP Header ----- TCP: Source port = 21 TCP: Destination port = 33185 </pre>
---	--

<i>Suite trame 39</i>	<i>Suite trame 40</i>
TCP: Sequence number = 192020144	TCP: Sequence number = 1347648250
TCP: Acknowledgement number = 1347648250	TCP: Acknowledgement number = 192020150
TCP: Data offset = 20 bytes	TCP: Data offset = 20 bytes
TCP: Flags = 0x18	TCP: Flags = 0x18
TCP: ..0. = No urgent pointer	TCP: ..0. = No urgent pointer
TCP: ...1 = Acknowledgement	TCP: ...1 = Acknowledgement
TCP: 1... = Push	TCP: 1... = Push
TCP:0.. = No reset	TCP:0.. = No reset
TCP:0. = No Syn	TCP:0. = No Syn
TCP:0 = No Fin	TCP:0 = No Fin
...	...
FTP: ----- FTP: -----	FTP: ----- FTP: -----
FTP: "QUIT\r\n"	FTP: "221 Goodbye.\r\n"
FTP:	FTP:

<i>trame 41</i>	<i>trame 42</i>
ETHER: ----- Ether Header -----	ETHER: ----- Ether Header -----
...	...
ETHER: Destination = 0:0:c:3a:4c:ce, Cisco	ETHER: Destination = 8:0:20:74:8c:a0, Sun
ETHER: Source = 8:0:20:74:8c:a0, Sun	ETHER: Source = 0:0:c:3a:4c:ce, Cisco
ETHER: Ethertype = 0800 (IP)	ETHER: Ethertype = 0800 (IP)
ETHER:	ETHER:
IP: ----- IP Header -----	IP: ----- IP Header -----
...	...
IP: Source address = 194.199.229.99, gtr-server1	IP: Source address = 193.49.108.10, simone.lirmm.fr
IP: Destination address= 193.49.108.10,simone.lirmm.fr	IP: Destination address = 194.199.229.99, gtr-server1
IP: No options	IP: No options
IP:	IP:
TCP: ----- TCP Header -----	TCP: ----- TCP Header -----
TCP: Source port = 33185	TCP: Source port = 21
TCP: Destination port = 21 (FTP)	TCP: Destination port = 33185
TCP: Sequence number = 192020150	TCP: Sequence number = 1347648264
TCP: Acknowledgement number = 1347648264	TCP: Acknowledgement number = 192020150
TCP: Data offset = 20 bytes	TCP: Data offset = 20 bytes
TCP: Flags = 0x11	TCP: Flags = 0x11
TCP: ..0. = No urgent pointer	TCP: ..0. = No urgent pointer
TCP: ...1 = Acknowledgement	TCP: ...1 = Acknowledgement
TCP: 0... = No push	TCP: 0... = No push
TCP:0.. = No reset	TCP:0.. = No reset
TCP:0. = No Syn	TCP:0. = No Syn
TCP:1 = Fin	TCP:1 = Fin
...	...
FTP: ----- FTP: -----	FTP: ----- FTP: -----
FTP: ""	FTP: ""
FTP:	FTP:

<i>trame 43</i>	
ETHER: ----- Ether Header -----	
...	
ETHER: Destination = 0:0:c:3a:4c:ce, Cisco	
ETHER: Source = 8:0:20:74:8c:a0, Sun	
ETHER: Ethertype = 0800 (IP)	
ETHER:	
IP: ----- IP Header -----	
...	
IP: Source address = 194.199.229.99, gtr-server1	
IP: Destination address= 193.49.108.10,simone.lirmm.fr	
IP: No options	
IP:	
TCP: ----- TCP Header -----	
TCP: Source port = 33185	
TCP: Destination port = 21 (FTP)	
TCP: Sequence number = 192020151	
TCP: Acknowledgement number = 1347648265	
TCP: Data offset = 20 bytes	
TCP: Flags = 0x10	
TCP: ..0. = No urgent pointer	
TCP: ...1 = Acknowledgement	
TCP: 0... = No push	
TCP:0.. = No reset	
TCP:0. = No Syn	
TCP:0 = No Fin	
...	
FTP: ----- FTP: -----	
FTP: ""	
FTP:	

TD2 Etude du modèle TCP/IP

L'objectif est d'étudier le *cheminement des trames* à travers plusieurs réseaux interconnectés. L'étude du cheminement des trames nous amènera au problème de *routing* de datagrammes.

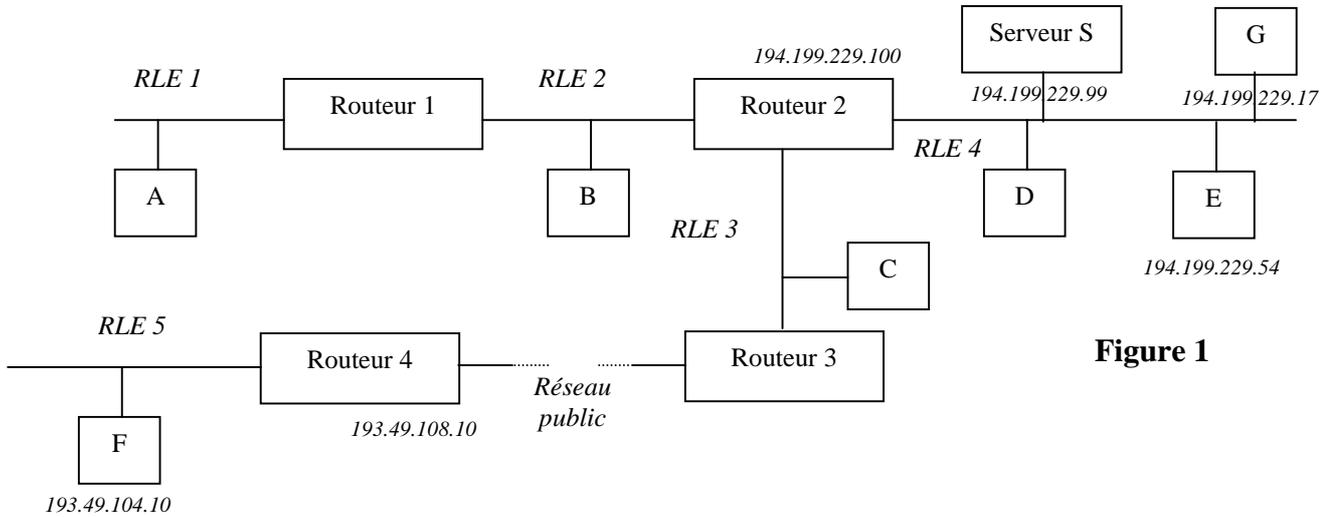


Figure 1

Résolution de nom et d'adresse

1) Etudier la trame ci-dessous pour extraire les adresses physiques et les adresses IP des machines concernées. Quels sont les domaines concernés ? Les machines concernées par ces adresses IP correspondent-elles à celles désignées par les adresses physiques ? Pourquoi ?

<pre>ETHER: ----- Ether Header ----- ETHER: Packet 1 arrived at 17:33:20.05 ETHER: Packet size = 58 bytes ETHER: Destination = 08:00:c3a:4c:ce, Cisco ETHER: Source = 8:0:20:74:8c:a0, Sun ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- IP: Version = 4 IP: Header length = 20 bytes IP: Type of service = 0x00 IP: xxx. = 0 (precedence) IP: ...0 = normal delay IP: 0... = normal throughput IP: 0.. = normal reliability IP: Total length = 44 bytes IP: Identification = 59245 IP: Flags = 0x4 IP: .1.. = do not fragment IP: ..0. = last fragment IP: Fragment offset = 0 bytes IP: Time to live = 255 seconds/hops IP: Protocol = 6 (TCP) IP: Header checksum = bef6 IP: Source address = 194.199.229.99, gtr-server1 IP: Destination address=193.49.108.10,simone.lirmm.fr IP: No options</pre>	<pre>TCP: ----- TCP Header ----- TCP: Source port = 33185 TCP: Destination port = 21 (FTP) TCP: Sequence number = 192020081 TCP: Acknowledgement number = 0 TCP: Data offset = 24 bytes TCP: Flags = 0x02 TCP: ..0. = No urgent pointer TCP: ...0 = No acknowledgement TCP: 0... = No push TCP: 0.. = No reset TCP: 1. = Syn TCP: 0 = No Fin TCP: Window = 8760 TCP: Checksum = 0x14ee TCP: Urgent pointer = 0 TCP: Options: (4 bytes) TCP: - Maximum segment size = 1460 bytes TCP: FTP: ----- FTP: ----- FTP: " " FTP:</pre>
--	--

<u>Nom</u>	<u>IP Address</u>	<u>Phys Addr</u>
simone.lirmm.fr	193.49.108.10	08:00:07:01:ba:c8
r_gtr.iutbeziers.univ-montp2.fr	194.199.229.100	00:00:0c:3a:4c:ce
gtr-server2.iutbeziers.univ-montp2.fr	194.199.229.54	00:80:5f:cc:8f:59
gtr-server1.iutbeziers.univ-montp2.fr	194.199.229.99	08:00:20:74:8c:a0
gtr-machine.iutbeziers.univ-montp2.fr	194.199.229.17	00:20:af:59:fc:28

2) La station D souhaite ouvrir une connexion avec la station E (figure 1). Elle ne connaît que le nom de la station E : qu'elles sont les informations manquantes pour parvenir à constituer entièrement la trame jusqu'au niveau d'ethernet ? Où sont obtenues ces informations et à l'aide de quels protocoles ?

3) Etudier les trames ci-dessous et en déduire l'opération réalisée (même si les données échangées ne figurent pas) ainsi que les protocoles utilisés aux différents niveaux. Préciser le RLE dans lequel la trame a été saisie.

<i>Trame 1</i>	<i>Trame 2</i>
ETHER: ----- Ether Header ----- ETHER: Packet 1 arrived at 16:59:45.48 ETHER: Packet size = 74 bytes ETHER: Destination = 0:0:c:3a:4c:ce, Cisco ETHER: Source = 8:0:20:74:8c:a0, Sun ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- IP: Version = 4 IP: Header length = 20 bytes IP: Type of service = 0x00 IP: xxx. = 0 (precedence) IP: ...0 = normal delay IP: 0... = normal throughput IP:0.. = normal reliability IP: Total length = 60 bytes IP: Identification = 10755 IP: Flags = 0x4 IP: .1.. = do not fragment IP: ..0. = last fragment IP: Fragment offset = 0 bytes IP: Time to live = 255 seconds/hops IP: Protocol = 17 (UDP) IP: Header checksum = 8046 IP: Source address = 194.199.229.99,gtr-server1 IP: Destination address=193.49.104.10,lirmm.lirmm.fr IP: No options IP: UDP: ----- UDP Header ----- UDP: Source port = 53 UDP: Destination port = 53 (DNS) UDP: Length = 40 UDP: Checksum = FD3F UDP: DNS: ----- DNS: ----- DNS: "" DNS:	ETHER: ----- Ether Header ----- ETHER: Packet 2 arrived at 16:59:45.52 ETHER: Packet size = 106 bytes ETHER: Destination = 8:0:20:74:8c:a0, Sun ETHER: Source = 0:0:c:3a:4c:ce, Cisco ETHER: Ethertype = 0800 (IP) ETHER: IP: ----- IP Header ----- IP: Version = 4 IP: Header length = 20 bytes IP: Type of service = 0x00 IP: xxx. = 0 (precedence) IP: ...0 = normal delay IP: 0... = normal throughput IP:0.. = normal reliability IP: Total length = 92 bytes IP: Identification = 11440 IP: Flags = 0x0 IP: .0.. = may fragment IP: ..0. = last fragment IP: Fragment offset = 0 bytes IP: Time to live = 52 seconds/hops IP: Protocol = 17 (UDP) IP: Header checksum = 887a IP: Source address = 193.49.104.10,lirmm.lirmm.fr IP: Destination address=194.199.229.99,gtr-server1 IP: No options IP: UDP: ----- UDP Header ----- UDP: Source port = 53 UDP: Destination port = 53 (DNS) UDP: Length = 72 UDP: Checksum = 0000 (no checksum) UDP: DNS: ----- DNS: ----- DNS: "" DNS: DNS:

4) Etudier les trames ci-dessous et en déduire l'opération réalisée et son mode de diffusion, ainsi que le protocole utilisé. Dans quel RLE a été saisie cette trame ? La trame aurait pu être directement adressée à la machine ayant l'adresse physique 08 00 20 74 8c a0. Pourquoi ?

<i>Trame 1</i>	<i>Trame 2</i>
Adresse destination = FF FF FF FF FF FF	Adresse destination = 00 20 af 59 fc 28
Adresse source = 00 20 af 59 fc 28	Adresse source = 08 00 20 74 8c a0
Type de datagramme = 08 06	type de datagramme = 08 06
Type de matériel = 00 01	type de matériel = 00 01
Type de protocole = 08 00	type de protocole = 08 00
Taille matériel = 06	taille matériel = 06
Taille protocole = 04	taille protocole = 04
Op = 00 01	Op = 00 02
Adresse source = 00 20 af 59 fc 28	Adresse source = 08 00 20 74 8c a0
Adresse IP source = 194 199 229 17	Adresse IP source = 194 199 229 99
Adresse cible = 00 00 00 00 00 00	Adresse cible = 00 20 af 59 fc 28
Adresse IP cible = 194 199 229 99	Adresse IP cible = 194 199 229 17

Routage statique et dynamique

A partir de la topologie représentée figure 1, traiter les scénarii suivants:

- 1) Le routeur 1 reçoit un paquet du RLE 2 pour A : où le transmet-il ? Connait-il RLE1 et RLE2 ? Où est inscrite cette information ? Quels sont, pour le routeur 1, les octets significatifs de l'adresse IP du RLE 1 ?
- 2) Le routeur 1 reçoit un paquet du RLE 2 pour B : qu'en fait-il ?
- 3) Le routeur 1 reçoit un paquet du RLE 1 pour F : qu'en fait-il ? Quelle est la route suivie par ce paquet ?
- 4) Si le routeur 2 doit router un paquet sans avoir trouvé d'information relative à cette destination dans sa table de routage, quelle est l'adresse IP de destination utilisée ?
- 5) Pour le routage dynamique les routeurs (les machines en charge de routage) testent leurs états ou échangent des informations par le biais de protocoles ICMP (Internet Control Message Protocol) et RIP (Routing Information Protocol).
Décoder l'échange que constitue les trames suivantes.

<i>Trame 1</i>	<i>Trame 2</i>
Adresse destination = 00 a0 24 47 0c f3	Adresse destination = 08 00 20 74 8c a0
Adresse source = 08 00 20 74 8c a0	Adresse source = 00 a0 24 47 0c f3
type de datagramme = 08 00	type de datagramme = 08 00
version + longueur en tête = 45	version + longueur en tête = 45
type de service = 00	type de service = 00
longueur totale = 00 54	longueur totale = 00 54
identification = c2 df	identification = 06 9e
flag + fragment offset = 40 00	flag + fragment offset = 00 00
durée de vie = ff	durée de vie = 40
protocole = 01	protocole = 01
somme de contrôle = 68 a1	somme de contrôle = 23 e4
Adresse IP source = 194 199 229 99	Adresse IP source = 194 199 229 52
Adresse IP destination = 194 199 229 52	Adresse IP destination = 194 199 229 99
Type = 08	Type = 00
code = 00	code = 00
somme de contrôle = 1f e9	somme de contrôle = 27 e9

6) Voici une illustration d'un échange de table de routage entre un routeur actif (propagation des infos) et un routeur passif (écoute et mise à jour seulement). Ces échanges sont périodiques (environ toutes les 30sec). Quel est le routeur actif ?

Le protocole utilisé, RIP, est basé sur le vecteur de distances où la notion de distance désigne le nombre de routeurs à traverser pour atteindre la destination. Construire la nouvelle table de routage du routeur K à partir des données du routeur J sachant que :

- Une route n'est modifiée dans une table que si une nouvelle route est indiquée ou la distance d'une route existante est modifiée,
- Si deux routes équivalentes sont indiquées par des messages RIP, c'est la première information qui est prise en compte,
- Pour un routeur, à chaque route de sa table est associée une tempo qui est réinitialisée à chaque message RIP indiquant que la route existe toujours.

<i>table du routeur K</i>		
<i>destination</i>	<i>distance</i>	<i>route</i>
réseau 1	0	directe
réseau 2	0	directe
réseau 4	8	routeur L
réseau 17	5	routeur M
réseau 24	6	routeur J
réseau 30	2	routeur Q
réseau 42	2	routeur J

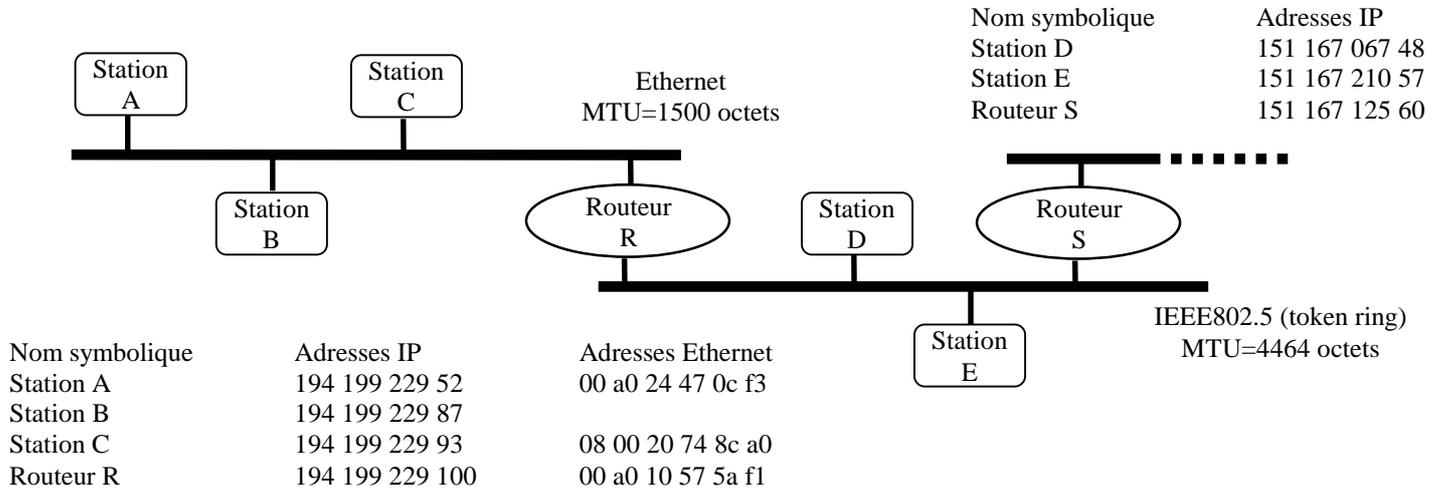
<i>Données du routeur J</i>	
<i>Destination</i>	<i>Distance</i>
Réseau 1	2
Réseau 4	3
Réseau 17	6
Réseau 21	4
Réseau 24	6
Réseau 30	10
Réseau 42	3

<i>Nouvelle table du routeur K</i>		
<i>destination</i>	<i>distance</i>	<i>route</i>

TD3 Etude de la couche RESEAU (modèle TCP/IP)

L'objectif est d'encapsuler des données issues de la couche TRANSPORT dans un datagramme IP puis de transmettre ce datagramme à la couche LIENS. Cette dernière doit alors encapsuler le datagramme dans une trame ETHERNET.

La figure ci-dessous décrit la topologie du réseau considéré.

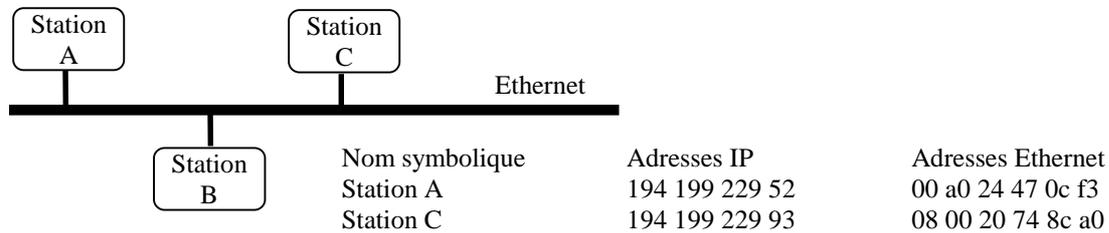


- Les données de la couche TRANSPORT (protocole TCP) ont une longueur de 300 octets. Construire le datagramme IP à transmettre à la couche LIENS DE DONNEES sachant que :
 - l'échange a lieu entre la Station A et la Station C,
 - la version courante du protocole IP est la version 4,
 - le type de service est supposé être service normal,
 - la durée de vie est fixée à 32.
 - l'identificateur (en hexa) du datagramme précédent était : 37 F1.
- La station A ne connaît pas l'adresse physique (adresse ethernet) de la station C. Décrire les échanges qui se produisent avant la transmission du datagramme IP (nature des échanges, sous-réseau et stations concernés, adresses ethernet).
- Construire la trame ETHERNET dans laquelle est encapsulé le datagramme IP précédent.
- L'échange précédent a lieu maintenant entre la Station A et la Station E. Quels sont les champs du datagramme IP et/ou de la trame ETHERNET affectés et leurs nouvelles valeurs ?
- Reprenons l'échange entre la Station A et la Station C. On suppose maintenant que les données de la couche TRANSPORT (protocole TCP) ont une longueur de 3000 octets. Décrire comment est (théoriquement) constitué le(s) datagramme(s) IP transmis vers la couche LIENS. Quels sont les champs du datagramme IP affectés et leurs nouvelles valeurs ?
- Considérons l'échange de 3 Ko de données de la Station E vers la Station A. Les deux sous-réseaux concernés sont respectivement des réseaux IEEE802.5 et ETHERNET. Vu leur MTU respectif, que se passe-t-il au niveau du routeur ?
- Que se passerait-il dans le cas d'un échange inverse (ie de A vers E) ?

TD4 Etude de la couche TRANSPORT (modèle TCP/IP)

L'objectif est d'étudier l'acheminement des informations non plus de machine à machine, comme le fait le protocole IP, mais **de bout en bout** c'est-à-dire de processus en processus.

La figure ci-dessous décrit la topologie du réseau considéré.



A- Connexion et mécanisme de port

Décrire par l'intermédiaire d'un diagramme les différentes étapes de l'établissement d'une connexion TCP entre la station A (ouverture active) et la station C (ouverture passive). Donner les extrémités de connexion (appelées parfois *socket*), et indiquer les champs affectés par l'ouverture de connexion et leurs valeurs.

- L'application (couche supérieure) concernée est FTP. Le numéro de port éphémère du client est 3217.
- La station A représente le client et la station C le serveur.
- La station A numérote ses segments TCP à partir de 200 et la station C à partir de 500.

B- Le transfert de données et la fenêtre d'acquittement

Décrire par l'intermédiaire d'un diagramme l'échange de données entre les stations A et C. Construire les en-tête TCP pour chaque trame échangée.

On suppose dans un premier temps que le transfert s'effectue sans problème.

- L'échange comprend 3 trames de données d'une longueur supposée de 300 octets, et une dernière trame de 200 octets.
- La **taille de la fenêtre** est de 300 octets et la **fenêtre glissante** (acquittement) est de taille 2.

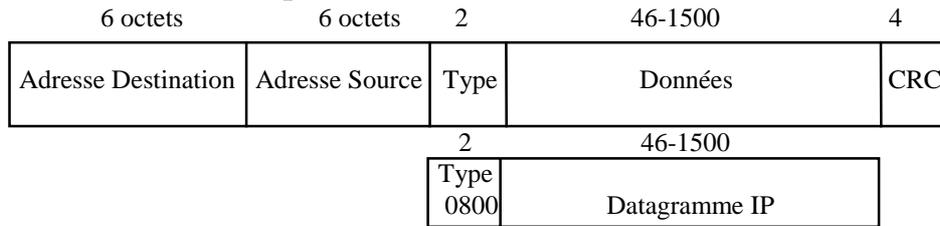
On suppose maintenant que l'accusé de réception de la deuxième trame, émis par la station A, s'est perdu (donc jamais reçu par la station C). Expliquer comment est détectée la non-réception de la trame perdue, et décrire le déroulement des échanges.

C- La fin d'une connexion

Décrire par l'intermédiaire d'un diagramme les différentes étapes de la fermeture de connexion TCP entre la station A et la station C. Indiquer les champs affectés par la fermeture de connexion et leurs valeurs.

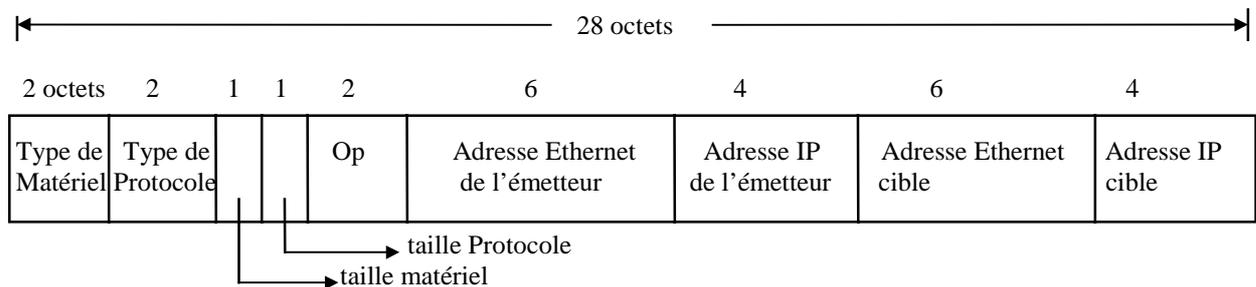
Format des en-têtes de trame, datagramme et segment

Encapsulation ETHERNET (RFC 894)



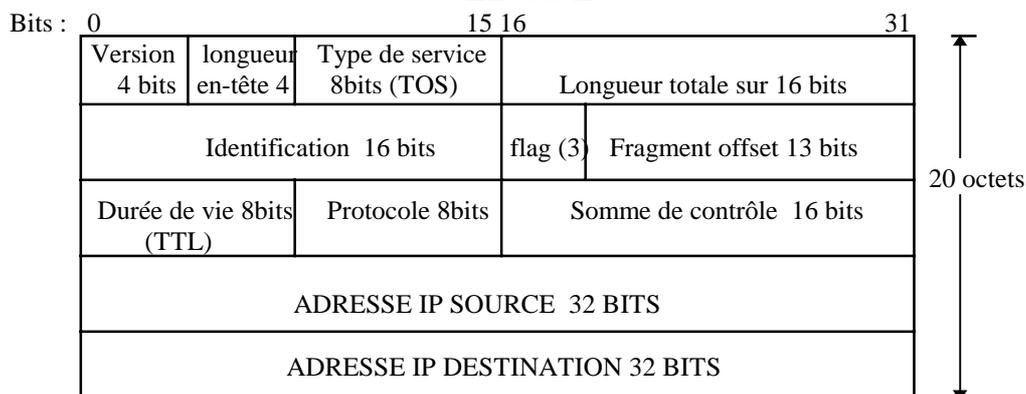
- Les *adresses* sont les adresses Ethernet en hexadécimal.
- *Type* : type de datagramme qui va suivre (0x0800 : IP ; 0x0806 : ARP)
- *CRC* : code de détection d'erreur (Cyclic Redondancy Check).

Format du paquet ARP



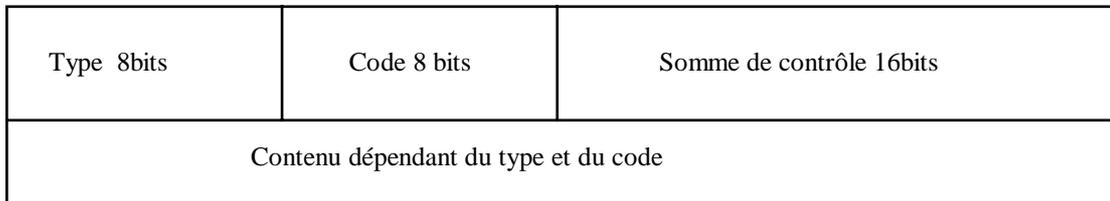
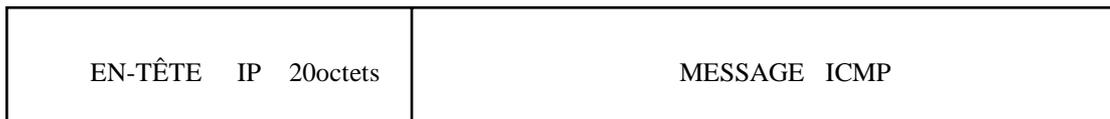
- *Type de matériel* : 1 pour une réseau Ethernet
- *Type de Protocole*: 0x0800 si ce sont des adresses IP indiquées dans la suite du message
- *Taille Matériel et Protocole* : indique la taille en octets des adresses matérielle et de protocole qui vont suivre.
- *Op* : indique si l'opération est une requête ARP (question, Op=1), une réponse ARP (Op=2), une question RARP (Op=3) ou une réponse RARP (Op=4).

En-tête IP



- *version* : version courante de IP (normalement 4).
- *longueur en-tête* : c'est le nombre de mots de 32 bits figurant dans l'en-tête.
- *TOS* : dépend des applications utilisées, de la forme 000xyzw0 avec x = minimise le délai ; y = maximalise le débit ; z = maximalise la fiabilité ; w = minimise le coût monétaire. Seul un des 4 bits est à 1, ou tous à 0 dans le cas d'un service normal.
- *Identification* : identificateur unique pour chaque datagramme IP. Incrémenté de 1 à chaque fois.
- *Longueur totale* : Contient la taille totale en octets du datagramme IP.
- *flag* : spécifie si les données transmises sont fragmentées (000 fragmentation autorisée, dernier fragment ; 001 fragmentation autorisée, pas dernier fragment ; 010 fragmentation pas autorisée).
- *fragment offset* : position du fragment (comptée en octets) par rapport au début des données (datagr. initial).
- *TTL (time-to-live)* : nombre maximum de routeurs que le datagramme peut traverser (souvent 32 ou 64).
- *Protocole* : indique le protocole du datagramme ou du segment suivant ; 1 si ICMP, 6 si TCP.
- *Somme de contrôle* : somme du contenu des paires d'octets de l'en-tête IP.

MESSAGE ICMP



- *Type* : ce champ est susceptible de prendre 15 valeurs , il indique quel est le type de message de contrôle.
- *Code* : le code est utilisé pour préciser dans quel cadre est émis le message de contrôle.

Exemple : type : 0 , code : 0 → Réponse écho (réponse de Ping)
 type : 8 , code : 0 → Demande écho (question de Ping)