Nom: Checa Hernàndez, Martí
Rocha Guzmán, Alejandra Lisette

P2 - Sessió 2: Estudi dels protocols TCP/IP en xarxa Ethernet commutada

Estudi del protocol IP en una xarxa Ethernet commutada

Es tracta de comprovar empíricament que totes les nostres comunicacions es converteixen en un flux de paquets IP amb adreça d'origen o destinació la del nostre PC. També s'observarà l'encapsulat de diferents tipus de tràfic (World Wide Web i ping). Finalment, es constatarà que en una xarxa commutada no es rebran trames/paquets dirigits a altres dispositius.

Per això, es generarà tràfic mitjançant ping o navegació Web alhora que es captura tota la informació que s'envia i rep mitjançant Wireshark.

Wireshark és un analitzador de protocols que permet veure tot el tràfic que es rep o s'envia per una interfície de xarxa d'un ordinador. En el nostre cas, aquesta targeta serà Ethernet. La informació que mostra s'organitza en 4 zones, que de dalt a baix són (veure Figura 1):

- **Filtre**: perquè només es mostrin aquelles trames Ethernet que compleixen amb la condició del filtre. El filtre pot ser el nom d'un protocol (per exemple, "ip"), el valor d'un camp d'un protocol (per exemple "ip.address==147.83.2.3") o combinacions de dues o més condicions ("&&" o AND i "||" o OR).

- **Resum de les trames capturades**: ordenades per un índex i una marca temporal, es pot observar l'adreça d'origen i destinació, així com informació relativa al seu contingut.

- **Encapsulat**: permet entrar en el contingut de la trama capturada, protocol per protocol i camp per camp, mitjançant menús desplegables.

- **Codificació**: seleccionant un camp amb el ratolí, es pot veure la seva codificació en hexadecimal.

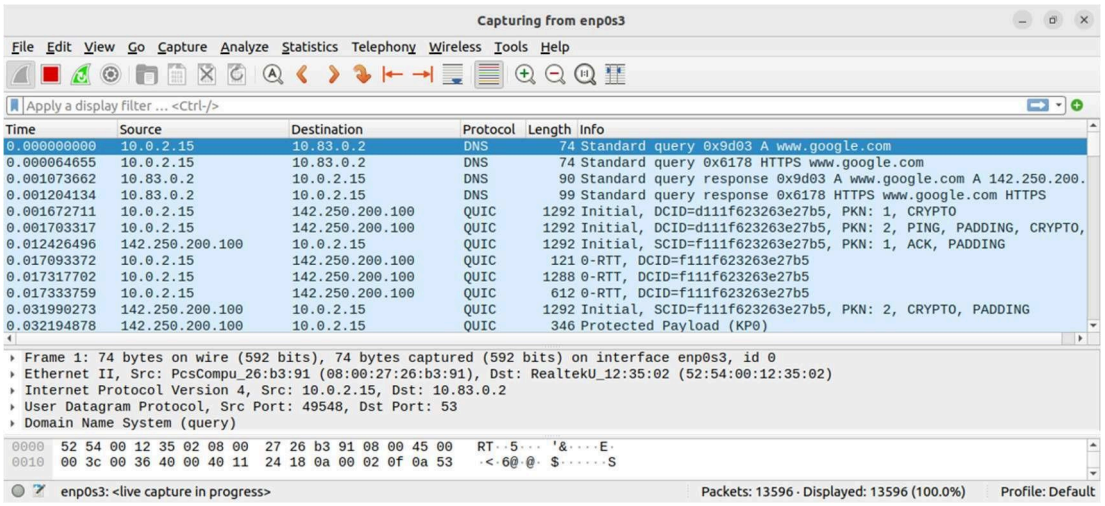
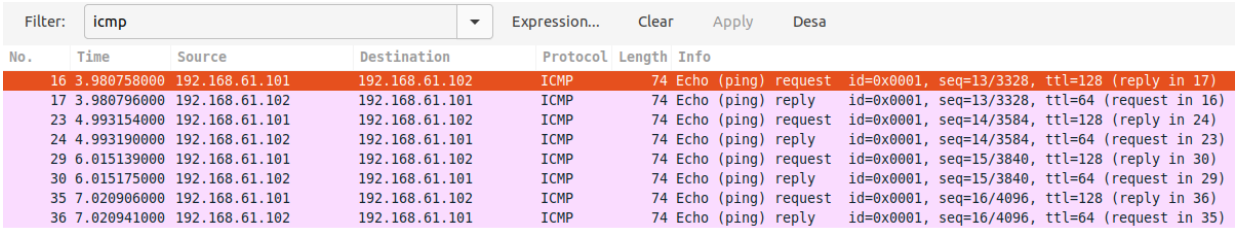


Figura 1. Analizador de protocolos Wireshark.

Tasques:

1. Comprovar si tots els paquets que s'envien i es reben tenen com a adreça d'origen l'adreça del PC



No.	Time	Source	Destination	Protocol	Length	Info
16	3.980758000	192.168.61.101	192.168.61.102	ICMP	74	Echo (ping) request id=0x0001, seq=13/3328, ttl=128 (reply in 17)
17	3.980796000	192.168.61.102	192.168.61.101	ICMP	74	Echo (ping) reply id=0x0001, seq=13/3328, ttl=64 (request in 16)
23	4.993154000	192.168.61.101	192.168.61.102	ICMP	74	Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 24)
24	4.993190000	192.168.61.102	192.168.61.101	ICMP	74	Echo (ping) reply id=0x0001, seq=14/3584, ttl=64 (request in 23)
29	6.015139000	192.168.61.101	192.168.61.102	ICMP	74	Echo (ping) request id=0x0001, seq=15/3840, ttl=128 (reply in 30)
30	6.015175000	192.168.61.102	192.168.61.101	ICMP	74	Echo (ping) reply id=0x0001, seq=15/3840, ttl=64 (request in 29)
35	7.020906000	192.168.61.101	192.168.61.102	ICMP	74	Echo (ping) request id=0x0001, seq=16/4096, ttl=128 (reply in 36)
36	7.020941000	192.168.61.102	192.168.61.101	ICMP	74	Echo (ping) reply id=0x0001, seq=16/4096, ttl=64 (request in 35)

Com es pot observar a l'hora de fer els pings desde l'ordinador Windows al de Linux, l'adreça d'origen del ping (en aquest cas 192.168.61.101) es manté igual en tots els camps pertinents per a cada request i reply que es genera. Pero no tots els paquets capturats han de ser necessàriament del PC amb el que estem treballant, ja que podem rebre en qualsevol moment qualsevol altre tipus de tràfic que s'hagi enviat al nostre ordinador, ja sigui per broadcast o directament.

2. Contrastar la resta de camps de la capçalera IP amb els vistos en teoria

```
▾ Internet Protocol Version 4, Src: 147.83.140.18 (147.83.140.18), Dst: 192.168.61.102 (192.168.61.102)
    Version: 4
    Header Length: 20 bytes
  ▾ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
    Total Length: 136
    Identification: 0x4934 (18740)
  ▾ Flags: 0x00
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 127
    Protocol: UDP (17)
  ▾ Header checksum: 0xd4bc [validation disabled]
      [Good: False]
      [Bad: False]
    Source: 147.83.140.18 (147.83.140.18)
    Destination: 192.168.61.102 (192.168.61.102)
```

El millor exemple que podem donar és la part IP de qualsevol paquet. Podem observar que tots els camps que hem estudiat i les seves respectives mides són iguals a les quals observem d'un paquet real.

3. Comprovar si s'envien o reben altres tipus de tràfic que no siguin paquets IP

Encara que tinguem altres protocols i serveis per sobre de la capa de xarxa, en tots casos hauran de ser encapsulats per un paquet IP per poder ser enviats. Pero com a tal, podem tindre paquets que no siguin només paquets del protocol TCP/IP o que estiguin per sobre d'ell.

4. Realitzar filtres per visualitzar els paquets d'una determinada adreça IP o protocol

Com es pot comprovar per les captures dels exercicis, podem modificar els filtres de manera molt versàtil per buscar paquets molt específics. Ja sigui per la seva adreça física, lògica, de destí, d'origen, per protocol, etc…

5. Descriure l'encapsulat (protocols) d'un ping i dels paquets enviats i rebuts al navegar per la Web

```
▾ Ethernet II, Src: 10:e7:c6:2c:91:f6 (10:e7:c6:2c:91:f6), Dst: 10:e7:c6:1c:2c:6f (10:e7:c6:1c:2c:6f)
  ▾ Destination: 10:e7:c6:1c:2c:6f (10:e7:c6:1c:2c:6f)
      Address: 10:e7:c6:1c:2c:6f (10:e7:c6:1c:2c:6f)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
  ▾ Source: 10:e7:c6:2c:91:f6 (10:e7:c6:2c:91:f6)
      Address: 10:e7:c6:2c:91:f6 (10:e7:c6:2c:91:f6)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IP (0x0800)
▾ Internet Protocol Version 4, Src: 192.168.61.101 (192.168.61.101), Dst: 192.168.61.102 (192.168.61.102)
    Version: 4
    Header Length: 20 bytes
  ▾ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... ..00 = Explicit Congestion Notification: Not-ECT (Not ECN-Capable Transport) (0x00)
    Total Length: 60
    Identification: 0x0fc3 (4035)
  ▾ Flags: 0x00
      0... .... = Reserved bit: Not set
      .0.. .... = Don't fragment: Not set
      ..0. .... = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: ICMP (1)
  ▾ Header checksum: 0x2ee2 [validation disabled]
      [Good: False]
      [Bad: False]
    Source: 192.168.61.101 (192.168.61.101)
    Destination: 192.168.61.102 (192.168.61.102)
▾ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4d4a [correct]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 17 (0x0011)
    Sequence number (LE): 4352 (0x1100)
    [Response frame: 19]
  ▸ Data (32 bytes)
```

L'encapsulat d'un paquet ping és molt senzill. Tenim dades "brosa" que enviem per omplir el camp de dades. Seguim amb el missatge ICMP que s'encarrega de declarar que el tipus de missatge a enviar és per donar servei al ping. Continuem amb el paquet IP que no requereix de molta informació ja que volem enviar un missatge molt simple a un altre ordinador. Finalment la trama Ethernet també molt senzilla només declara desde quin dispositiu enviem la sol·licitud i a quin l'enviem.

6. Anàlisi de les captures: mapeig entre adreces IP i MAC

```
e9501366@aul-1924:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.60.134           ether   c4:34:6b:5b:47:d3   C                     eno1
192.168.61.104           ether   10:e7:c6:19:3d:46   C                     eno1
192.168.61.85            ether   10:e7:c6:31:c0:5e   C                     eno1
192.168.61.210           ether   10:e7:c6:31:c1:1a   C                     eno1
192.168.60.252           ether   18:a9:05:ba:1b:f0   C                     eno1
192.168.60.2             ether   00:1b:a9:62:04:80   C                     eno1
192.168.60.172           ether   6c:62:6d:5a:26:24   C                     eno1
192.168.61.101           ether   10:e7:c6:2c:91:f6   C                     eno1
192.168.60.241           ether   00:24:81:96:ec:b5   C                     eno1
192.168.61.254           ether   00:09:0f:a7:b3:32   C                     eno1
192.168.61.252           ether   1a:92:27:a9:37:24   C                     eno1
192.168.60.83            ether   50:65:f3:50:8e:04   C                     eno1
```

Connectivitat amb un PC del laboratori

Comenceu a capturar tràfic amb Wireshark al PC Linux i envieu un ping des del PC Windows al PC Linux (ping adreça_IP_Linux).

a) A Wireshark, apliqueu el filtre icmp perquè mostri únicament els paquets ICMP. Quins tipus de paquets s'observen? Quina és l'adreça IP d'origen i destinació en cadascun d'ells?

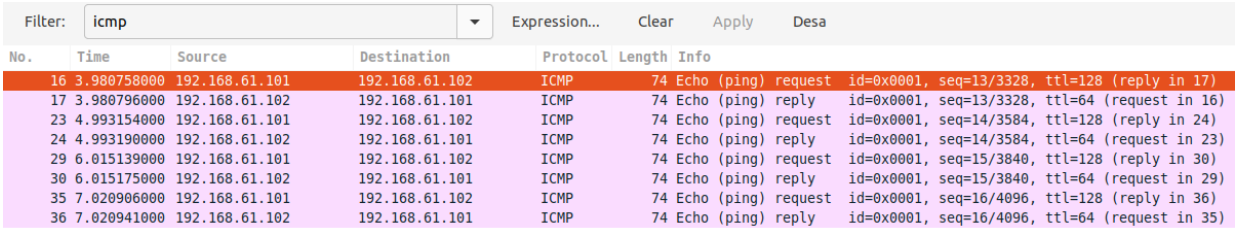
Ordinador Linux:

```
e9501366@aul-1924:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
p default qlen 1000
    link/ether 10:e7:c6:1c:2c:6f brd ff:ff:ff:ff:ff:ff
    altname enp0s31f6
    inet 192.168.61.102/23 brd 192.168.61.255 scope global noprefixroute eno1
       valid_lft forever preferred_lft forever
    inet6 fe80::d6bc:512b:f690:f99b/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Ping desde l'ordinador Windows:

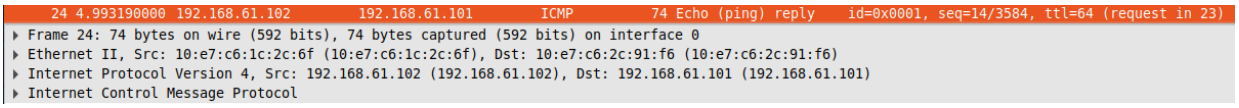
```
C:\Users\f5160909>ping 192.168.61.102

Haciendo ping a 192.168.61.102 con 32 bytes de datos:
Respuesta desde 192.168.61.102: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.61.102: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.61.102: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.61.102: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.61.102:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Wireshark:



Observem que tots els paquets dels protocol ICMP que s'han enviat són part de la comanda ping que s'ha fet desde l'ordinador de Windows. Per defecte, Windows envia 4 pings a l'adreça esmentada, i a Wireshark podem observar els 4 ping requests de l'ordinador de Windows i les 4 replies de l'ordinador de Linux. En ambdós casos podem observar que l'adreça de l'ordinador de Windows és 192.168.61.101 i la de Linux és 192.168.61.102.

b) Quina versió d'IP s'utilitza en tots els paquets?



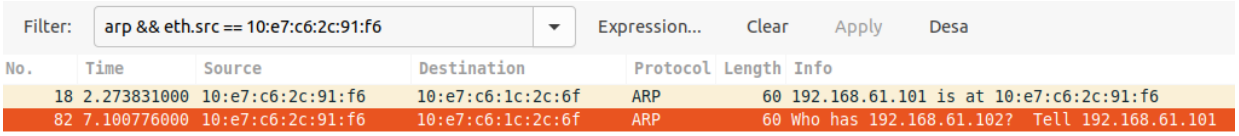
Podem observar que tots els paquets del protocol ICMP de tipus 8 (ping) estan utilitzant la versió 4 de IP. Això ho podem veure tant per la informació dels headers del paquet, com el protocol en si ja que ICMP utilitza un protocol específic per a connexions o requests que utilitzin IPv6 amb ICMPv6.

c) Indiqueu quina és l'estructura de la trama Ethernet que encapsula aquests paquets. Per a cada capçalera, indiqueu el valor dels camps més rellevants (MAC origen i destinació, IP origen i destinació, protocol)

	MAC Origen	IP Origen	MAC Destí	IP Destí	Protocol
Request (Windows)	10:E7:C6: 2C:91:F6	192.168. 61.101	10:E7:C6: 1C:2C:6F	192.168. 61.102	IP
Reply (Linux)	10:E7:C6: 1C:2C:6F	192.168. 61.102	10:E7:C6: 2C:91:F6	192.168. 61.101	IP

Encara que haguem enviat 4 pings desde l'ordinador de Windows, els principals camps seran els mateixos entre els Replies i els Requests. L'únic que variarà serà on estan situades les adreces de destí i les adreces d'origen. El protocol en tots casos ha de ser el mateix.

d) Canvieu el filtre a Wireshark a arp perquè mostri els paquets ARP. Busqueu els paquets ARP en els quals l'adreça MAC d'origen sigui la del PC Windows o la del PC Linux. Per a què serveixen?



Els paquets ARP que s'han enviat serveixen per establir una connexió entre els dos dispositius. Estan relacionant la IP lògica amb la seva adreça física. Aquest establiment només es farà un cop, a menys que la taula ARP s'ompli completament i el Sistema Operatiu la borri o l'usuari esborri aquest establiment de la taula manualment. En el nostre cas, està "desordenat" perquè l'ordinador Windows ja tenia el dispositiu a la seva taula. En canvi, hem esborrat l'establiment de la taula ARP de l'ordinador de Linux i ha hagut de restablir-ho.

e) Visualitzeu la taula ARP del PC Linux amb la comanda arp -n (si està buida, torneu a executar el ping i la comanda arp). Quines adreces s'observen? Quina relació té el contingut de la taula amb els paquets observats a l'apartat d)?

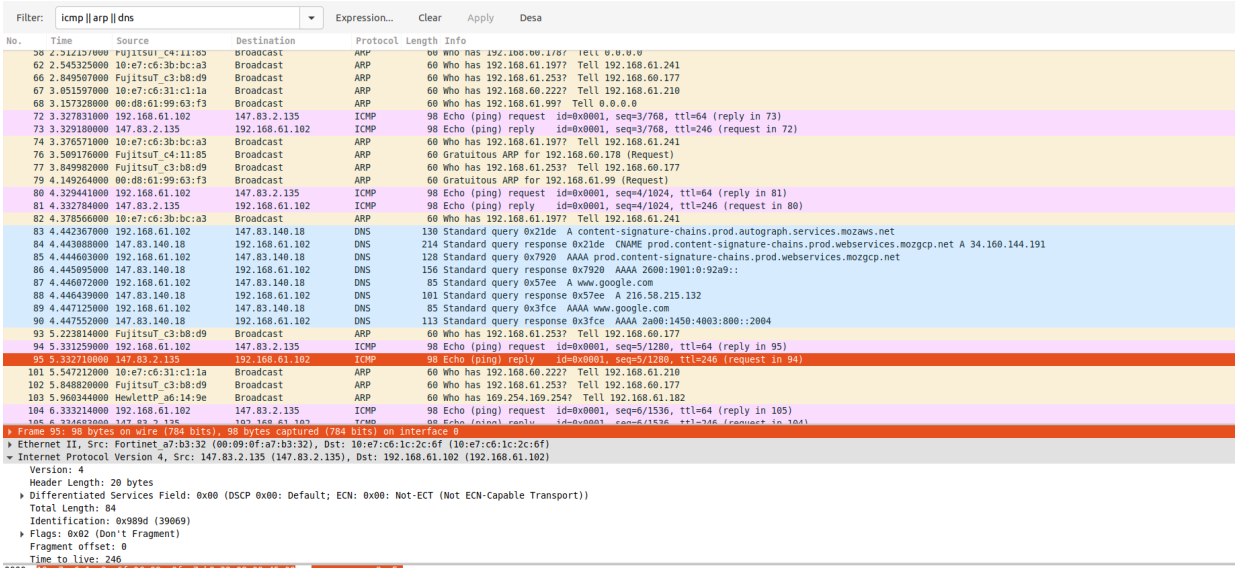
```
e9501366@aul-1924:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.60.134           ether   c4:34:6b:5b:47:d3   C                     eno1
192.168.61.104           ether   10:e7:c6:19:3d:46   C                     eno1
192.168.61.85            ether   10:e7:c6:31:c0:5e   C                     eno1
192.168.61.210           ether   10:e7:c6:31:c1:1a   C                     eno1
192.168.60.252           ether   18:a9:05:ba:1b:f0   C                     eno1
192.168.60.2             ether   00:1b:a9:62:04:80   C                     eno1
192.168.60.172           ether   6c:62:6d:5a:26:24   C                     eno1
192.168.61.101           ether   10:e7:c6:2c:91:f6   C                     eno1
192.168.60.241           ether   00:24:81:96:ec:b5   C                     eno1
192.168.61.254           ether   00:09:0f:a7:b3:32   C                     eno1
192.168.61.252           ether   1a:92:27:a9:37:24   C                     eno1
192.168.60.83            ether   50:65:f3:50:8e:04   C                     eno1
```

A la taula ARP podem observar totes les adreces de la nostra xarxa LAN del laboratori amb la seva adreça física. Com que tothom està fent el laboratori i està enviant ARP requests per la IP de broadcast, tothom està rebent aquestes sol·licituds i afegint-hi els dispositius que van enviant els diferent paquets ARP. En el nostre cas, l'ordinador Windows que ens importa té l'adreça IPv4 192.168.61.101 i l'adreça MAC 10:E7:C6:2C:91:F6

f) Pareu de capturar tràfic i guardeu la captura amb el nom ping_linux. Adjunteu-la a l'entrega a Atenea.

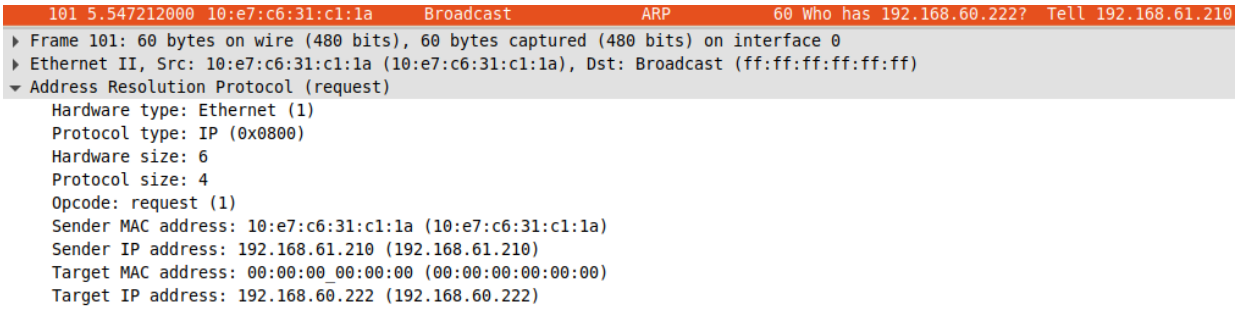
Connectivitat amb un host extern

a) Comenceu a capturar tràfic de nou (al PC Linux) i realitzeu un ping a www.upc.edu. Canvieu el filtre de Wireshark a icmp || arp || dns. Comproveu que tots els paquets utilitzen la mateixa versió que a l'apartat anterior. Analitzeu l'estructura de la trama Ethernet que conté els paquets ICMP. Quin valor tenen els principals camps? Han canviat respecte a l'apartat anterior – Connectivitat amb un PC del laboratori - ? Per què?



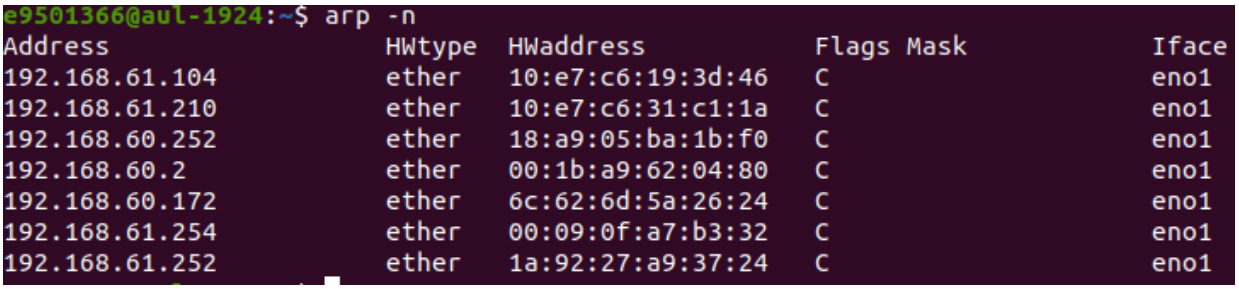
Tots els paquets dels protocols filtrats utilitzen IPv4 encara que tinguin la capacitat de treballar amb IPv6. La trama Ethernet per als paquets del protocol ICMP són molt senzills. Només tenen una adreça de Host, una adreça de destí, i el protocol al qual està donant servei. En el nostre cas, aquest protocol entra dins de l'especificació IP.

b) Analitzeu els paquets ARP enviats i rebuts pel PC Linux. Quina informació contenen?



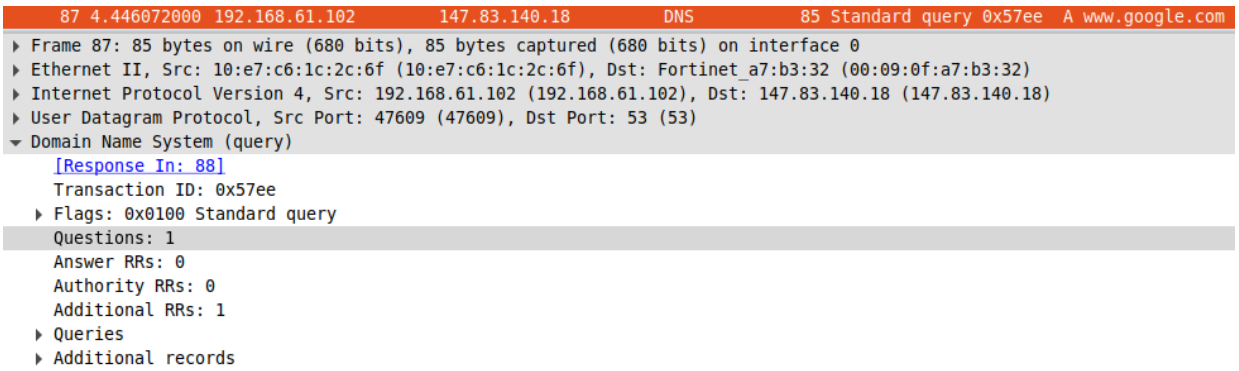
Els paquets ARP contenen la informació del medi per on s'està enviant les dades al nivel físic d'enllaç, el protocol superior al que està donant servei, el tamany de l'adreça la qual utilitza el medi d'enllaç (Una adreça MAC són 48 bits = 6 bytes), el tamany de l'adreça que utilitza el nivell superior (IPv4 són 32 bits = 4 bytes), el Opcode que indica si el paquet és de Request (1) o de Response (2), l'adreça MAC i IPv4 del dispositiu que envia, i l'adreça MAC i IPv4 del dispositiu que està intentant trobar.

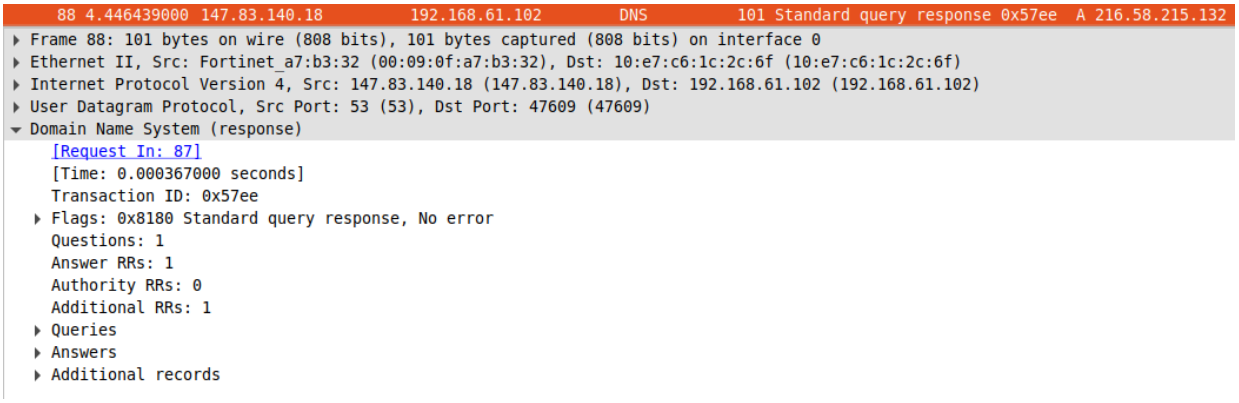
c) Visualitzeu la taula ARP del PC Linux amb la comanda arp -n (si està buida, torneu a executar el ping i la comanda arp). Quines adreces s'observen?



Totes les adreces de la taula formen part de la xarxa LAN del laboratori a la qual pertany el nostre dispositiu. Aquí s'han guardat totes les adreces que s'han anat consultant a l'hora de fer ARP requests que s'han guardat a la ARP table del nostre ordinador amb la seva MAC address per relacionar-les. D'aquesta manera es poden comunicar sense la necessitat de repetir un altre ARP Request.

d) Analitzeu els paquets DNS. Indiqueu quina és l'adreça IP d'origen i destinació dels mateixos. Per a què serveixen?

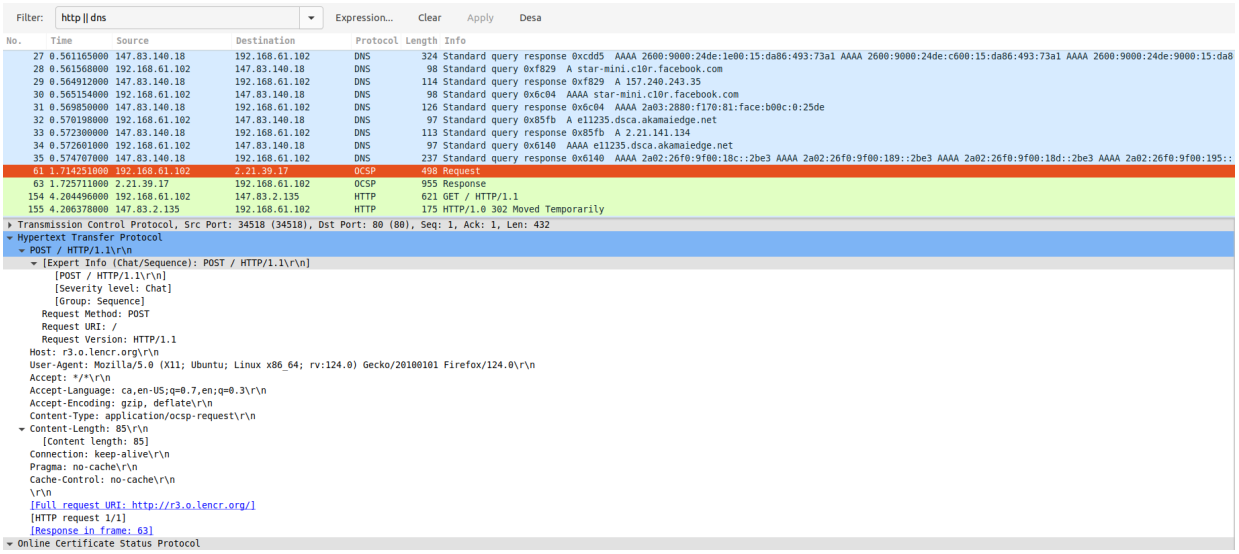




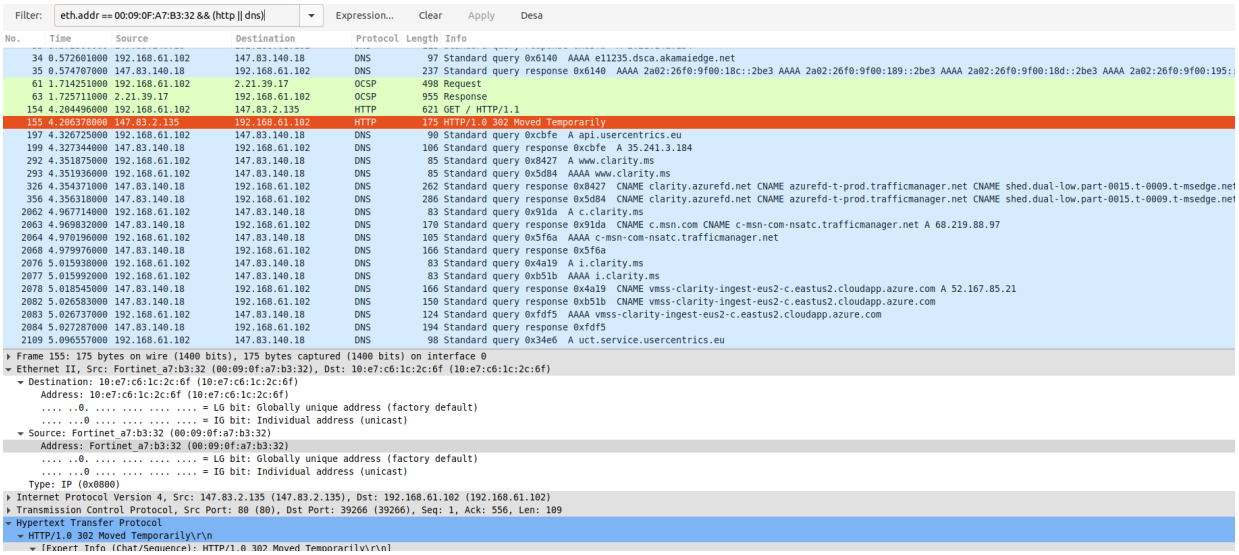
Els paquets DNS constent de dues parts, de manera similar a ARP. El primer fa una query a una adreça determinada. En aquest cas a www.google.com. Aquesta demana l'adreça IP del Host del hipervincle al qual s'està accedint a més de detalls com ara la llargada de l'adreça en caràcters o per tags. La resposta retorna l'adreça que el Host del hipervincle està assignat. En aquest cas, www.google.com està assignat a 216.58.215.132, una adreça de Classe A.

Per generar tràfic web, obriu el navegador i carregueu la web www.upc.edu. Canvieu el filtre a Wireshark a http || dns.

e) Analitzeu l'estructura de les trames que contenen els paquets http. Indiqueu tots els protocols que apareixen (i a quina capa del model TCP/IP pertanyen). Indiqueu també si el valor dels camps MAC d'origen i destinació, IP d'origen i destinació coincideixen amb els de l'apartat b).



Els protocols que apareixen són DNC, OSCP i HTTP. Tots pertanyen a la capa d'aplicació i utilitzen TCP a la capa de Xarxa ja que els serveis que ofereixen requereixen de la fiabilitat que TCP assegura, encara que els protocols HTTP puguin ser implementats amb UDP.



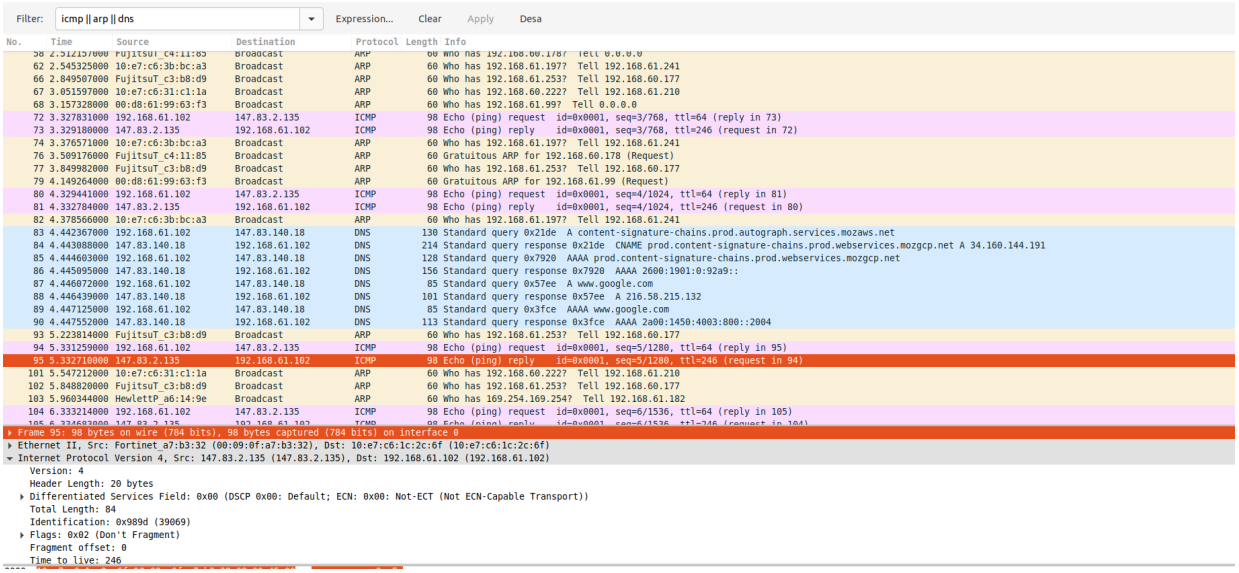
Amb el filtre **eth.addr == 00:09:0F:A7:B3:32 && (http || dns)** podem comprobar que l'adreça MAC que vam trobar a l'apartat b) utilitzant el servei ping assenyala la mateixa adreça que hem capturat a l'hora de entrar a la pàgina manualment.

f) Pareu de capturar tràfic i guardeu la captura amb el nom ping_extern. Adjunteu-la a l'entrega a Atenea.

Finalment, torneu a capturar paquets amb Wireshark durant uns minuts, pareu la captura i respongueu a la següent pregunta:

g) S'observa algun paquet amb adreça IP de destinació diferent a la del PC? Si la resposta és sí, raoneu si concorda amb el fet d'utilitzar una xarxa Ethernet commutada.

Utilitzarem la captura que hem generat als exercicis anteriors per raonar la resposta a aquesta pregunta.



El millor exemple que podem utilitzar per comprovar que podem rebre paquets amb una adreça de destinació diferent a la nostre és gràcies al protocol ARP. El protocol ARP envia una sol·licitud broadcast a la xarxa a la qual pertany el client demanant una adreça en específic. Encara que no siguem aquella adreça que demana, com que la sol·licitud s'ha enviat en multicast i l'adreça de destí és la que demana

(no necessàriament la nostre), és possible que es doni aquest cas. Això concorda amb el fet d'utilitzar una xarxa Ethernet conmutada ja que demostra la connexió que hi ha entre tots els dispositius de la xarxa local i com aquesta sol·licitud broadcast s'ha rebut per tots els dispositius actualment connectats.

h) Indiqueu el filtre que hauríeu d'utilitzar a Wireshark per visualitzar únicament els paquets amb IP de destinació la del vostre PC.

El filtre que hauríem utilitzat a Wireshark és ip.dst == 192.168.61.102. Si volguessim buscar per trames ethernet en comptes d'adreces IP utilitzariem eth.dst == <Adreça MAC>