

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

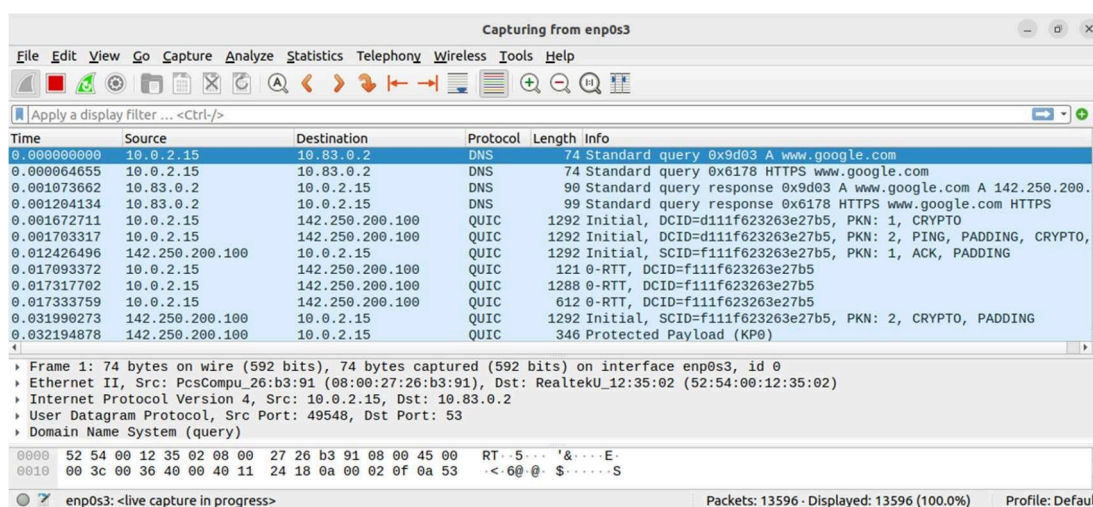


Figura 1. Analizador de protocolos Wireshark.

Tasques:

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

Filter:	icmp			Expression...	Clear	Apply	Desa
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 default 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 group 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:

Filter:	icmp			Expression...	Clear	Apply	Desa
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)	

Observem que tots els paquets del 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?

```

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)
  Frame 24: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
  Ethernet II, Src: 10:e7:c6:1c:2c:6f (10:e7:c6:1c:2c:6f), Dst: 10:e7:c6:2c:91:f6 (10:e7:c6:2c:91:f6)
  Internet Protocol Version 4, Src: 192.168.61.102 (192.168.61.102), Dst: 192.168.61.101 (192.168.61.101)
  Internet Control Message Protocol
  
```

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?

Filter:	arp && eth.src == 10:e7:c6:2c:91:f6	▼	Expression...	Clear	Apply	Desa
No.	Time	Source	Destination	Protocol	Length	Info
18	2.273831000	10:e7:c6:2c:91:f6	10:e7:c6:1c:2c:6f	ARP	60	192.168.61.101 is at 10:e7:c6:2c:91:f6
82	7.100776000	10:e7:c6:2c:91:f6	10:e7:c6:1c:2c:6f	ARP	60	Who has 192.168.61.102? Tell 192.168.61.101

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 diferents 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?

No.	Time	Source	Destination	Protocol	Length	Info
58	2.21219000	Fujitsu c4:11:80	broadcast	ARP	60	Who has 192.168.60.178? Tell 0.0.0.0
62	2.545325000	10:e7:c6:3b:bc:a3	Broadcast	ARP	60	Who has 192.168.61.197? Tell 192.168.61.241
66	2.849507000	Fujitsu c3:b8:d9	Broadcast	ARP	60	Who has 192.168.61.253? Tell 192.168.60.177
67	3.051597000	10:e7:c6:31:c1:1a	Broadcast	ARP	60	Who has 192.168.60.227? Tell 192.168.61.210
68	3.137328000	00:d8:61:99:63:f3	Broadcast	ARP	60	Who has 192.168.61.99? Tell 0.0.0.0
72	3.327831000	192.168.61.102	147.83.2.135	ICMP	98	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 73)
73	3.329180000	147.83.2.135	192.168.61.102	ICMP	98	Echo (ping) reply id=0x0001, seq=3/768, ttl=246 (request in 72)
74	3.376571000	10:e7:c6:3b:bc:a3	Broadcast	ARP	60	Who has 192.168.61.197? Tell 192.168.61.241
76	3.509170000	Fujitsu c4:11:80	Broadcast	ARP	60	Gratuitous ARP for 192.168.60.178 (Request)
77	3.849982000	Fujitsu c3:b8:d9	Broadcast	ARP	60	Who has 192.168.61.253? Tell 192.168.60.177
79	4.149264000	00:d8:61:99:63:f3	Broadcast	ARP	60	Gratuitous ARP for 192.168.61.99 (Request)
80	4.329441000	192.168.61.102	147.83.2.135	ICMP	98	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (reply in 81)
81	4.332784000	147.83.2.135	192.168.61.102	ICMP	98	Echo (ping) reply id=0x0001, seq=4/1024, ttl=246 (request in 80)
82	4.378560000	10:e7:c6:3b:bc:a3	Broadcast	ARP	60	Who has 192.168.61.197? Tell 192.168.61.241
83	4.442367000	192.168.61.102	147.83.140.18	DNS	130	Standard query 0x21de A content-signature-chains.prod.autograph.services.mozilla.net
84	4.443888000	147.83.140.18	192.168.61.102	DNS	214	Standard query response 0x21de CNAME prod.content-signature-chains.prod.webservices.mozilla.net A 34.160.144.191
85	4.444603000	192.168.61.102	147.83.140.18	DNS	128	Standard query 0x7920 AAAA prod.content-signature-chains.prod.webservices.mozilla.net
86	4.445095000	147.83.140.18	192.168.61.102	DNS	156	Standard query response 0x7920 AAAA 2600:1901:0:92a9::
87	4.446072000	192.168.61.102	147.83.140.18	DNS	85	Standard query 0x37ee A www.google.com
88	4.446439000	147.83.140.18	192.168.61.102	DNS	101	Standard query response 0x37ee A 216.58.215.132
89	4.447125000	147.83.140.18	192.168.61.102	DNS	85	Standard query 0x3fce AAAA www.google.com
90	4.447552000	147.83.140.18	192.168.61.102	DNS	113	Standard query response 0x3fce AAAA 2a00:1450:4003:800::2004
93	5.223814000	Fujitsu c3:b8:d9	Broadcast	ARP	60	Who has 192.168.61.253? Tell 192.168.60.177
94	5.331259000	192.168.61.102	147.83.2.135	ICMP	98	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (reply in 95)
95	5.332710000	147.83.2.135	192.168.61.102	ICMP	98	Echo (ping) reply id=0x0001, seq=5/1280, ttl=246 (request in 94)
101	5.547212000	10:e7:c6:31:c1:1a	Broadcast	ARP	60	Who has 192.168.60.227? Tell 192.168.61.210
102	5.848820000	Fujitsu c3:b8:d9	Broadcast	ARP	60	Who has 192.168.61.253? Tell 192.168.60.177
103	5.960344000	Hewlett a6:14:9e	Broadcast	ARP	60	Who has 169.254.169.254? Tell 192.168.61.182
104	6.333214000	192.168.61.102	147.83.2.135	ICMP	98	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (reply in 105)
105	6.334662000	147.83.2.135	192.168.61.102	ICMP	98	Echo (ping) reply id=0x0001, seq=6/1536, ttl=246 (request in 104)

Frame 95: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0

Ethernet II, Src: Fortinet a7:b3:32 (08:09:0f:a7:b3:32), Dst: 10:e7:c6:1c:2c:6f (10:e7:c6:1c:2c:6f)

Internet Protocol Version 4, Src: 147.83.2.135 (147.83.2.135), 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))

Total Length: 84

Identification: 0x989d (39069)

Flags: 0x02 (Don't Fragment)

Fragment offset: 0

Time to live: 246

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?

```
101 5.547212000 10:e7:c6:31:c1:1a Broadcast ARP 60 Who has 192.168.60.222? Tell 192.168.61.210
▶ Frame 101: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▶ Ethernet II, Src: 10:e7:c6:31:c1:1a (10:e7:c6:31:c1:1a), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▼ Address Resolution Protocol (request)
  Hardware type: Ethernet (1)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: 10:e7:c6:31:c1:1a (10:e7:c6:31:c1:1a)
  Sender IP address: 192.168.61.210 (192.168.61.210)
  Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.60.222 (192.168.60.222)
```

Els paquets ARP contenen la informació del medi per on s'està enviant les dades al nivell 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?

```
e9501366@aul-1924:~$ arp -n
Address HWtype HWaddress Flags Mask Iface
192.168.61.104 ether 10:e7:c6:19:3d:46 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
192.168.61.254 ether 00:09:0f:a7:b3:32 C eno1
192.168.61.252 ether 1a:92:27:a9:37:24 C eno1
```

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?

```
87 4.446072000 192.168.61.102 147.83.140.18 DNS 85 Standard query 0x57ee A www.google.com
▶ Frame 87: 85 bytes on wire (680 bits), 85 bytes captured (680 bits) on interface 0
▶ Ethernet II, Src: 10:e7:c6:1c:2c:6f (10:e7:c6:1c:2c:6f), Dst: Fortinet_a7:b3:32 (00:09:0f:a7:b3:32)
▶ Internet Protocol Version 4, Src: 192.168.61.102 (192.168.61.102), Dst: 147.83.140.18 (147.83.140.18)
▶ User Datagram Protocol, Src Port: 47609 (47609), Dst Port: 53 (53)
▼ Domain Name System (query)
  [Response In: 88]
  Transaction ID: 0x57ee
  Flags: 0x0100 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 1
  ▶ Queries
  ▶ Additional records
```

```
88 4.446439000 147.83.140.18 192.168.61.102 DNS 101 Standard query response 0x57ee A 216.58.215.132
▶ Frame 88: 101 bytes on wire (808 bits), 101 bytes captured (808 bits) on interface 0
▶ Ethernet II, Src: Fortinet_a7:b3:32 (00:09:0f:a7:b3:32), Dst: 10:e7:c6:1c:2c:6f (10:e7:c6:1c:2c:6f)
▶ Internet Protocol Version 4, Src: 147.83.140.18 (147.83.140.18), Dst: 192.168.61.102 (192.168.61.102)
▶ User Datagram Protocol, Src Port: 53 (53), Dst Port: 47609 (47609)
▼ Domain Name System (response)
  [Request In: 87]
  [Time: 0.000367000 seconds]
  Transaction ID: 0x57ee
  ▶ Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 1
  ▶ Queries
  ▶ Answers
  ▶ Additional records
```

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).

```
Filter: http || dns
Expression... Clear Apply Desa

No. Time Source Destination Protocol Length Info
27 0.561105800 147.83.140.18 192.168.61.102 DNS 324 Standard query response 0xcd5 AAAA 2600:9000:24de:1e00:15:da86:493:73a1 AAAA 2600:9000:24de:c600:15:da86:493:73a1 AAAA 2600:9000:24de:9000:15:da86:493:73a1
28 0.561508000 192.168.61.102 147.83.140.18 DNS 98 Standard query 0xf629 A star-mini.c10r.facebook.com
29 0.564912000 147.83.140.18 192.168.61.102 DNS 114 Standard query response 0xf629 A 157.240.243.35
30 0.565154000 192.168.61.102 147.83.140.18 DNS 98 Standard query 0x6c94 AAAA star-mini.c10r.facebook.com
31 0.569850000 147.83.140.18 192.168.61.102 DNS 126 Standard query response 0x6c94 AAAA 2a03:2880:f170:81:face:b00c:0:25de
32 0.570198000 192.168.61.102 147.83.140.18 DNS 97 Standard query 0x85fb A e11235.dsca.akamaiedge.net
33 0.572308000 147.83.140.18 192.168.61.102 DNS 113 Standard query response 0x85fb A 2.21.141.134
34 0.572601000 192.168.61.102 147.83.140.18 DNS 97 Standard query 0x6140 AAAA e11235.dsca.akamaiedge.net
35 0.574707000 147.83.140.18 192.168.61.102 DNS 237 Standard query response 0x6140 AAAA 2a02:26f0:9f00:189::2be3 AAAA 2a02:26f0:9f00:18d::2be3 AAAA 2a02:26f0:9f00:195::2be3 AAAA 2a02:26f0:9f00:195::2be3
61 1.714251000 192.168.61.102 2.21.39.17 OCSP 498 Request
63 1.725711000 192.168.61.102 192.168.61.102 OCSP 955 Response
154 4.204496000 192.168.61.102 147.83.2.135 HTTP 621 GET / HTTP/1.1
155 4.206378000 147.83.2.135 192.168.61.102 HTTP 175 HTTP/1.0 302 Moved Temporarily
▶ Transmission Control Protocol, Src Port: 34518 (34518), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 432
▼ Hypertext Transfer Protocol
  POST / HTTP/1.1\r\n
  [Expert Info (Chat/Sequence): POST / HTTP/1.1\r\n]
  [POST / HTTP/1.1\r\n]
  [Severity level: Chat]
  [Group: Sequence]
  Request Method: POST
  Request URI: /
  Request Version: HTTP/1.1
  Host: r3.o.lencr.org\r\n
  User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:124.0) Gecko/20100101 Firefox/124.0\r\n
  Accept: */*\r\n
  Accept-Language: ca,en-US;q=0.7,en;q=0.3\r\n
  Accept-Encoding: gzip, deflate\r\n
  Content-Type: application/ocsp-request\r\n
  Content-Length: 85\r\n
  [Content Length: 85]
  Connection: keep-alive\r\n
  Pragma: no-cache\r\n
  Cache-Control: no-cache\r\n
  \r\n
  [Full request URI: http://r3.o.lencr.org/]
  [HTTP request 1/1]
  [Response in frame: 63]
▼ Online Certificate Status Protocol
```

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.

Filter: eth.addr == 00:09:0F:A7:B3:32 && (http dns)							Expression...	Clear	Apply	Desa
No.	Time	Source	Destination	Protocol	Length	Info				
34	0.572601000	192.168.61.102	147.83.140.18	DNS	97	Standard query 0x6140 AAAA e11235.dsca.akamaiedge.net				
35	0.574070000	147.83.140.18	192.168.61.102	DNS	237	Standard query response 0x6140 AAAA 2a02:26f0:9f00:18c::2be3 AAAA 2a02:26f0:9f00:189::2be3 AAAA 2a02:26f0:9f00:18d::2be3 AAAA 2a02:26f0:9f00:195::2be3				
61	1.714251000	192.168.61.102	2.21.39.17	OCSP	498	Request				
63	1.725710000	2.21.39.17	192.168.61.102	OCSP	955	Response				
154	4.204496000	192.168.61.102	147.83.2.135	HTTP	621	GET / HTTP/1.1				
155	4.206378000	147.83.2.135	192.168.61.102	HTTP	175	HTTP/1.0 302 Moved Temporarily				
197	4.326725000	192.168.61.102	147.83.140.18	DNS	90	Standard query 0xcbf6 A api.usercentrics.eu				
199	4.327344000	147.83.140.18	192.168.61.102	DNS	106	Standard query response 0xcbf6 A api.usercentrics.eu				
292	4.351875000	192.168.61.102	147.83.140.18	DNS	85	Standard query 0x8427 A www.clarity.ms				
293	4.351936000	192.168.61.102	147.83.140.18	DNS	85	Standard query 0x5d84 AAAA www.clarity.ms				
326	4.354371000	147.83.140.18	192.168.61.102	DNS	262	Standard query response 0x8427 CNAME clarity.azurefd.net CNAME azurefd-t-prod.trafficmanager.net CNAME shed.dual-low.part-0015.t-0009.t-msedge.net				
356	4.356318000	147.83.140.18	192.168.61.102	DNS	286	Standard query response 0x5d84 CNAME clarity.azurefd.net CNAME azurefd-t-prod.trafficmanager.net CNAME shed.dual-low.part-0015.t-0009.t-msedge.net				
2862	4.907714000	192.168.61.102	147.83.140.18	DNS	83	Standard query 0x91da A c.clarity.ms				
2863	4.908832000	147.83.140.18	192.168.61.102	DNS	170	Standard query response 0x91da CNAME c.msn.com CNAME c.msn-com-nsatc.trafficmanager.net A 68.219.88.97				
2864	4.970196000	192.168.61.102	147.83.140.18	DNS	105	Standard query 0x5f6a AAAA c.msn-com-nsatc.trafficmanager.net				
2868	4.979976000	147.83.140.18	192.168.61.102	DNS	166	Standard query response 0x5f6a				
2876	5.015938000	192.168.61.102	147.83.140.18	DNS	83	Standard query 0x4a19 A i.clarity.ms				
2877	5.015992000	192.168.61.102	147.83.140.18	DNS	83	Standard query 0xb51b AAAA i.clarity.ms				
2878	5.018545000	147.83.140.18	192.168.61.102	DNS	166	Standard query response 0x4a19 CNAME vmss-clarity-ingest-eus2-c.eastus2.cloudapp.azure.com				
2882	5.026583000	147.83.140.18	192.168.61.102	DNS	150	Standard query response 0xb51b CNAME vmss-clarity-ingest-eus2-c.eastus2.cloudapp.azure.com				
2883	5.026737000	192.168.61.102	147.83.140.18	DNS	124	Standard query 0xfdf5 AAAA vmss-clarity-ingest-eus2-c.eastus2.cloudapp.azure.com				
2884	5.027287000	147.83.140.18	192.168.61.102	DNS	194	Standard query response 0xfdf5				
2109	5.096557000	192.168.61.102	147.83.140.18	DNS	98	Standard query 0x3466 A uct.service.usercentrics.eu				
Frame 155: 175 bytes on wire (1400 bits), 175 bytes captured (1400 bits) on interface 0										
Ethernet II, Src: Fortinet_a7:b3:32 (00:09:0f:a7:b3:32), 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: Fortinet_a7:b3:32 (00:09:0f:a7:b3:32)										
Address: Fortinet_a7:b3:32 (00:09:0f:a7:b3:32)										
.....0. = LG bit: Globally unique address (factory default)										
.....0. = IG bit: Individual address (unicast)										
Type: IP (0x0800)										
Internet Protocol Version 4, Src: 147.83.2.135 (147.83.2.135), Dst: 192.168.61.102 (192.168.61.102)										
Transmission Control Protocol, Src Port: 80 (80), Dst Port: 39266 (39266), Seq: 1, Ack: 556, Len: 109										
Hypertext Transfer Protocol										
HTTP/1.0 302 Moved Temporarily\r\n										
Expert Info (Chat/Sequence): HTTP/1.0 302 Moved Temporarily\r\n										

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) Paireu 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.

Filter: icmp arp dns							Expression...	Clear	Apply	Desa
No.	Time	Source	Destination	Protocol	Length	Info				
38	2.312137000	Fujitsu c4:11:85	Broadcast	ARP	60	Who has 192.168.60.17? Tell 0.0.0.0				
62	2.545325000	10:e7:c6:3b:bc:a3	Broadcast	ARP	60	Who has 192.168.61.197? Tell 192.168.61.241				
66	2.849507000	Fujitsu c3:b8:d9	Broadcast	ARP	60	Who has 192.168.61.253? Tell 192.168.60.177				
67	3.051597000	10:e7:c6:31:c1:1a	Broadcast	ARP	60	Who has 192.168.60.227? Tell 192.168.61.210				
68	3.137328000	00:d8:61:99:63:f3	Broadcast	ARP	60	Who has 192.168.61.99? Tell 0.0.0.0				
72	3.327831000	192.168.61.102	147.83.2.135	ICMP	98	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (reply in 73)				
73	3.329180000	147.83.2.135	192.168.61.102	ICMP	98	Echo (ping) reply id=0x0001, seq=3/768, ttl=246 (request in 72)				
74	3.376571000	10:e7:c6:3b:bc:a3	Broadcast	ARP	60	Who has 192.168.61.197? Tell 192.168.61.241				
76	3.509176000	Fujitsu c4:11:85	Broadcast	ARP	60	Gratuitous ARP for 192.168.60.178 (Request)				
77	3.849820000	Fujitsu c3:b8:d9	Broadcast	ARP	60	Who has 192.168.61.253? Tell 192.168.60.177				
79	4.149264000	00:d8:61:99:63:f3	Broadcast	ARP	60	Gratuitous ARP for 192.168.61.99 (Request)				
80	4.329441000	192.168.61.102	147.83.2.135	ICMP	98	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (reply in 81)				
81	4.332784000	147.83.2.135	192.168.61.102	ICMP	98	Echo (ping) reply id=0x0001, seq=4/1024, ttl=246 (request in 80)				
82	4.378566000	10:e7:c6:3b:bc:a3	Broadcast	ARP	60	Who has 192.168.61.197? Tell 192.168.61.241				
83	4.442367000	192.168.61.102	147.83.140.18	DNS	139	Standard query 0x21de A content-signature-chains.prod.autograph.services.mozaws.net				
84	4.443888000	147.83.140.18	192.168.61.102	DNS	214	Standard query response 0x21de CNAME prod.content-signature-chains.prod.webservices.mozgcp.net A 34.160.144.191				
85	4.444603000	192.168.61.102	147.83.140.18	DNS	128	Standard query 0x7920 AAAA prod.content-signature-chains.prod.webservices.mozgcp.net				
86	4.445095000	147.83.140.18	192.168.61.102	DNS	156	Standard query response 0x7920 AAAA 2600:1901:0:92a9::				
87	4.446072000	192.168.61.102	147.83.140.18	DNS	85	Standard query 0x37ee A www.google.com				
88	4.446439000	147.83.140.18	192.168.61.102	DNS	101	Standard query response 0x37ee A 216.58.215.132				
89	4.447125000	192.168.61.102	147.83.140.18	DNS	85	Standard query 0x3fce AAAA www.google.com				
90	4.447552000	147.83.140.18	192.168.61.102	DNS	113	Standard query response 0x3fce AAAA 2a00:1450:4003:800::2004				
93	5.223814000	Fujitsu c3:b8:d9	Broadcast	ARP	60	Who has 192.168.61.253? Tell 192.168.60.177				
94	5.331259000	192.168.61.102	147.83.2.135	ICMP	98	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (reply in 95)				
95	5.332710000	147.83.2.135	192.168.61.102	ICMP	98	Echo (ping) reply id=0x0001, seq=5/1280, ttl=246 (request in 94)				
101	5.547212000	10:e7:c6:31:c1:1a	Broadcast	ARP	60	Who has 192.168.60.227? Tell 192.168.61.210				
102	5.848820000	Fujitsu c3:b8:d9	Broadcast	ARP	60	Who has 192.168.61.253? Tell 192.168.60.177				
103	5.960344000	Howlett a0:14:9e	Broadcast	ARP	60	Who has 169.254.169.254? Tell 192.168.61.182				
104	6.333214000	192.168.61.102	147.83.2.135	ICMP	98	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (reply in 105)				
105	6.334636000	147.83.2.135	192.168.61.102	ICMP	98	Echo (ping) reply id=0x0001, seq=6/1536, ttl=246 (request in 104)				
Frame 95: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0										
Ethernet II, Src: Fortinet_a7:b3:32 (00:09:0f:a7:b3:32), Dst: 10:e7:c6:1c:2c:6f (10:e7:c6:1c:2c:6f)										
Internet Protocol Version 4, Src: 147.83.2.135 (147.83.2.135), 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))										
Total Length: 84										
Identification: 0x989d (39069)										
Flags: 0x02 (Don't Fragment)										
Fragment offset: 0										
Time to live: 246										

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

(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>`