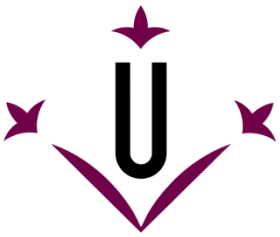


Universitat de Lleida

Escola Politècnica Superior

Grau en Enginyeria Informàtica

Xarxes

Pràctica 3

Anàlisi de trànsit

Serrano Ortega, Aniol

11/06/2023

GPraLab 3

Índex

Índex de taules

Índex d'il·lustracions

1. Introducció

L'objectiu d'aquesta pràctica és analitzar el trànsit d'una captura de xarxa mitjançant el programari de Wireshark. Aquesta és una eina de captura i anàlisi de paquets de xarxa que permet examinar detalladament el trànsit de dades i els protocols emprats.

2. Caracterització de la xarxa

A continuació es procedeix a elaborar una anàlisi general per tal d'obtenir una caracterització general de la xarxa. Mitjançant una sèrie de filtres i tècniques es poden determinar les característiques generals de la xarxa.

2.1. Tipus d'adreçament de la capa de xarxa

El tipus d'adreçament de la capa de xarxa fa referència al rang d'adreces IP utilitzades en una xarxa. Aquest adreçament es divideix en classes (A-E). Cada classe té un rang d'adreces assignat i una estructura específica per identificar les adreces de xarxa i amfitrió. En la següent taula es pot veure el rang d'adreces per a cada classe.

Taula 1: Rang d'adreçament segons el tipus de classe.

Classe	Rang d'adreces	Màscara
A	0.0.0.0 – 127.255.255.255	255.0.0.0
B	128.0.0.0 – 191.255.255.255	255.255.0.0
C	192.0.0.0 – 223.255.255.255	255.255.255.0
D	224.0.0.0 – 239.255.255.255	*Multicast*
E	240.0.0.0 – 255.255.255.255	Reservat

En la majoria dels paquets en el camp *Soruce* o en el *Destination* l'adreça és del tipus 172.16.X.X. Aquesta adreça està continguda dins del rang d'adreçament de la classe B. Per tant, es pot afirmar que la xarxa és de la classe B.

2.2. Adreça de xarxa

A partir de l'anàlisi realitzada prèviament en el tipus d'adreçament es pot veure que els paquets tant de l'emissor com del receptor son del tipus 172.16.X.X/16. Donat que l'adreça és del tipus B, els 16 primers bits romanen constants i 16 bits restants poden variar per a representar els diferents dispositius de la xarxa.

Amb aquest raonament es pot inferir que l'adreça de la xarxa és 172.16.0.0 i el rang d'adreçament IP és de 172.16.0.1 fins a 172.16.255.254. Per tant, l'adreça de xarxa és la 172.16.0.1/16. Les adreces 172.16.0.0 i 172.16.255.255 estan reservades, la primera serveix per a identificar la xarxa i la segona és l'adreça de *broadcast*.

2.3. Adreça de *broadcast*

Per determinar l'adreça de broadcast, tal com ja s'ha especificat prèviament, s'ha d'emprar la màscara de la xarxa de la classe B (255.255.0.0) i operar-la mitjançant operacions lògiques amb l'adreça de la xarxa (172.16.0.0).

Aplicant la negació lògica (NOT) sobre la màscara de la xarxa 255.255.0.0 s'obté 0.0.255.255. Seguidament, s'aplica l'operació OR entre l'adreça de la xarxa i el resultat i s'obté 172.16.255.255. Per tant, l'adreça 172.16.255.255 és l'adreça de *broadcast* per a la xarxa 172.16.0.0/16.

2.4. Adreça IPv4 de l'encaminador (*router)*

Primerament, per a identificar l'adreça IPv4 de la porta d'enllaç de la xarxa s'ha obtingut l'adreça MAC de l'encaminador amb el filtre `http`. És adequat analitzar un paquet *http* amb una adreça externa a la xarxa, ja que aquest ha de passar de forma obligatòria per l'encaminador i, per tant, es pot determinar la seva MAC.

Si s'analitza un altre tipus de paquet aquest no està garantit que passi per l'encaminador, la qual cosa pot donar a confusió. Aquesta adreça MAC es pot obtenir o bé en el camp *Source* o bé en el camp *Destination* en la secció d'*Ethernet II* en funció de si és un paquet d'anada o de tornada a l'encaminador.

La MAC obtinguda amb el filtre `http` és 00:16:36:8e:0f:9c. Mitjançant un paquet ARP[1] i la MAC obtinguda s'obté l'adreça IP amb el següent filtre:

```
arp && eth.addr == 00:16:36:8e:0f:9c
```

A l'analitzar un paquet emprant aquest filtre es pot veure en el camp de *Sender IP address* dins d'*Address Resolution Protocol (request)* que l'adreça de la porta d'enllaç de la xarxa és la 172.16.20.1.

2.5. Adreça IPv4 del *DNS*

A continuació es tracta d'esbrinar l'adreça del servidor de noms de la xarxa (DNS[2]). Per a mostrar únicament aquest tipus de paquets, s'ha aplicat el següent filtre:

```
dns
```

S'ha seleccionat un paquet del tipus *query* i s'ha seguit la seqüència UDP associada. En aquesta petició es pot veure en el camp *Destination*

[1] ARP (*Address Resolution Protocol*): Protocol per a traduir adreces IP a adreces MAC.

[2] DNS (*Domain Name System*): Sistema per a traduir noms de domini a adreces IP.

l'adreça IP del servidor de noms (DNS). La petició es fa a l'adreça 172.16.20.10, per tant, aquesta és l'adreça IP del servidor DNS.

Altrament, es pot emprar el mateix procediment però per a un paquet de resposta del servidor de noms. En conseqüència, l'adreça del servidor DNS es mostra en el camp *Source*.

3. Anàlisi nivells d'enllaç i de xarxa

A continuació, s'ha elaborat una anàlisi detallada dels nivells d'enllaç de dades (nivell 2) i de xarxa (nivell 3) per determinar els protocols encapsulats en les trames de nivell 2, així com les adreces i protocols presents en la xarxa.

3.1. Identificació dels protocols de trames de nivell 2

En la captura s'han identificat dos protocols de nivell 2, *Ethernet II*[3] i *IEEE 802.3 Ethernet*[4]. Aquests camps es poden observar en la mateixa informació de la trama de cada paquet. En general s'ha observat que tots els paquets usen el protocol *Ethernet II* de nivell 2 exceptuant els paquets LLC que usen *IEEE 802.3 Ethernet*.

Per a verificar aquesta informació, s'ha observat el protocol de nivell 2 usat per a cada protocol de nivell 3 emprat a cada paquet, i en cas de ser *Ethernet II* el protocol de nivell 3, s'ha exclòs mitjançant un filtre. A mesura que es filtren els protocols de nivell 3 que empren aquest protocol, s'ha observat que els únics paquets romanents han estat els LLC. Per tant, es pot afirmar que únicament aquests paquets empren el protocol *IEEE 802.3 Ethernet* i els restants empren el protocol *Ethernet II* de nivell 2 en aquesta captura.

[3] *Ethernet II*: Variant del protocol *Ethernet* de nivell 2.

[4] *IEEE 802.3 Ethernet*: Estàndard que especifica les característiques generals de la xarxa *Ethernet*.

El filtre resultant ha estat el següent:

```
!arp && !tcp && !udp && !icmpv6 && !vrrp && !igmp
```

A l'excloure els paquets que empren el mateix protocol es pot afegir el filtre `llc` per a veure aquells que empren el protocol *IEEE 802.3 Ethernet* de nivell 2. Un cop verificat que únicament hi ha dos estàndards de nivell 2 presents en la captura, es procedeix a indicar el valor del camp *type* de la trama de nivell 2.

Les PDU[5] de les trames de nivell 2 presents en la captura són les següents:

Taula 2: PDU de IEEE 802.3 Ethernet

Preàmbul	**SD[6]**	**Adreça destí**	**Adreça origen**	***Length/ Type***	**Dades**	**FCS[7]**
7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46 - 1500 bytes	4 bytes

Taula 3: PDU de Ethernet II

Preàmbul	**Adreça destí**	**Adreça origen**	***Type***	**Dades**	**FCS**
8 bytes	6 bytes	6 bytes	2 bytes	46 - 1500 bytes	4 bytes

En la PDU de *IEEE 802.3 Ethernet* es pot observar que el camp *type* és emprat tant per a definir la llargada de la trama, com per definir el tipus de protocol encapsulat de nivell 3. Si el contingut d'aquest camp és menor o igual a 1500 en decimal s'interpreta com la longitud del paquet. Altrament, si el valor del camp és major o igual a 1536 en decimal s'interpreta com a tipus de protocol.

[5] PDU (*Protocol Data Unit*): Unitat de dades que conté l'estructura de les capçaleres d'un protocol.

[6] SD (*Start Delimiter*): Bit per a permetre al receptor de la trama identificar i sincronitzar-se amb l'inici de la trama *Ethernet*.

[7] FCS (*Frame Check Sequence*): Camp pera verificar la integritat de la trama.

En el cas d'*Ethernet II* el camp *type* també és de 2 bytes, però el tipus d'informació no depèn del contingut. D'igual manera, aquest camp també indica el protocol encapsulat de nivell 3.

El tipus de la trama de nivell 2 en cada paquet es pot observar en els detalls de la trama en el camp *Ethernet II* o *IEEE 802.3 Ethernet* el camp *Type* o *Length/Type*. En aquests camps es poden observar tots els protocols de nivell 3 encapsulats. A partir d'aquesta informació s'ha elaborat la següent taula:

Taula 4: Protocols de nivell 3 encapsulats en la trama de nivell 2

Nivell 2	Nivell 3	***Type* o *Length*/Type**
Ethernet II	IPv4	0x0800
Ethernet II	IPv6	0x86dd
Ethernet II	ARP	0x0806
Ethernet II	IPX / SPX	0x8137
IEEE 802.3 Ethernet	LLC	—

Per a verificar que aquests són els unics protocols s'ha usat el següent filtre per a descartar els protocols ja revisats:

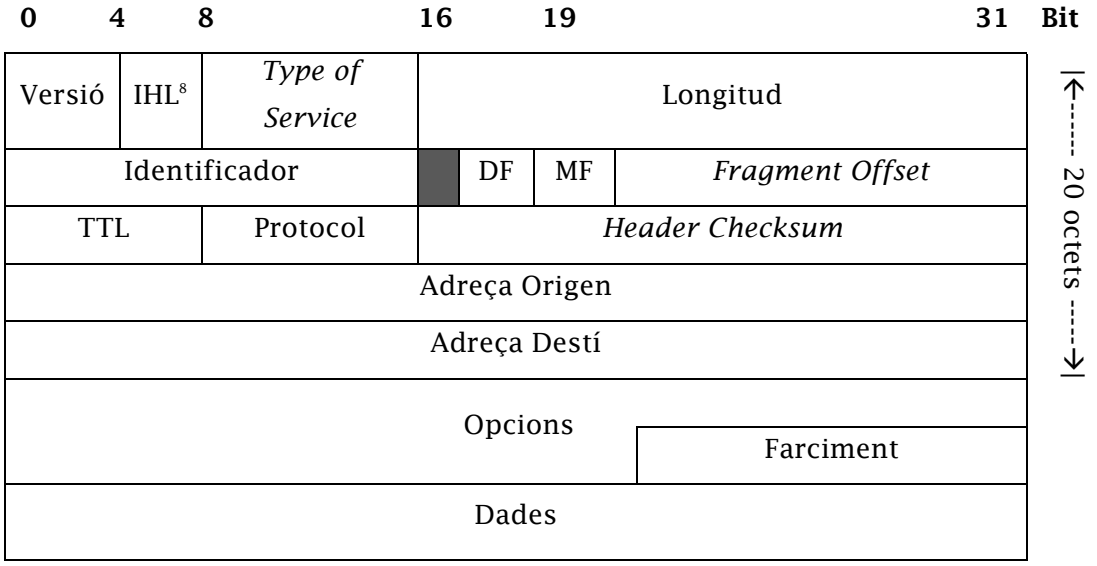
`eth.type == <Type>`

A continuació es mostren les PDU en bits dels protocols de nivell 3 mostrats en la taula 4.

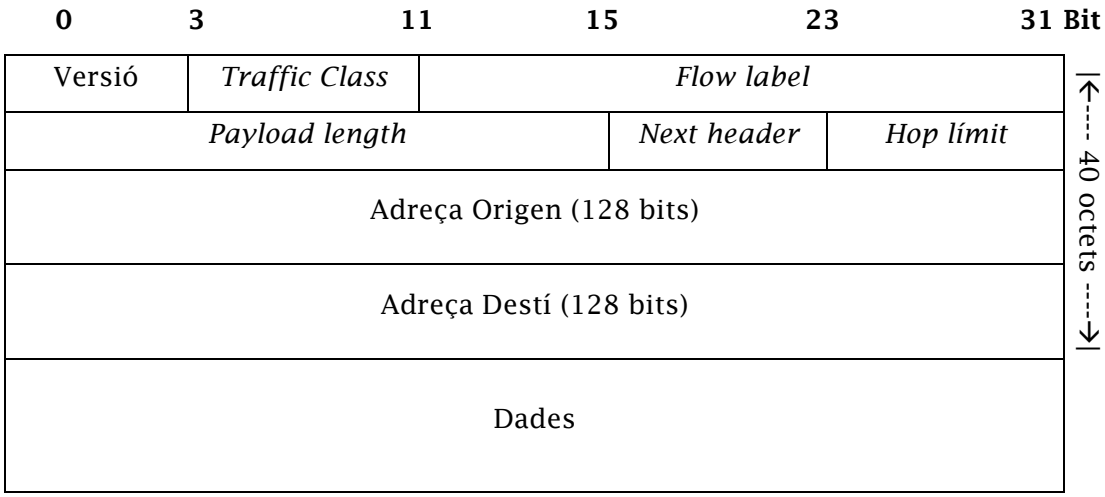
Taula 5: PDU de IPX / SPX

16 bits	**16 bits**	**8 bits**	**8 bits**	**32 bits**
Checksum	Llargada	*Transport Control*	*Packet Type*	*Destination Network*
48 bits	**16 bits**	**32 bits**	**48 bits**	**16 bits**
Destination Node	*Destination Socket*	*Source Network*	*Source Node*	*Source Socket*

Taula 6: PDU de IPv4



Taula 7: PDU de IPv6



Taula 8: PDU de LLC

8 bits	8 bits	8 o 16 bits	Variable
DSAP	SSAP	Control	Dades

[8] IHL (*Internet Header Length*): Longitud, en paraules de 32 bit, de la capçalera.

Taula 9: PDU de ARP

0	8	16	31
Hardware Type		*Protocol Type*	
HAL[9]	*PAL*[10]	Operació	
SHA (Bytes 0-3)			
Source Hardware Address (4-5 Bytes)		*Source Protocol Address* (0-1 Bytes)	
Source Protocol Address (2-3 Bytes)		*Target Hardware Address* (0-1 Bytes)	
Target Hardware Address *(Bytes 2-5)*			
Target Protocol Address *(Bytes 0-3)*			

3.2. Equips que empren els protocols IPX i IPv4

Per a identificar els paquets que empren el protocol IPX[11] s'usa el següent filtre:

```
ipx
```

Mitjançant aquest filtre es mostren únicament els paquets que empren aquest protocol. A partir d'aquests paquets s'han extret les adreces MAC de tots els paquets. Per a verificar que efectivament no hi ha adreces MAC repetides s'ha seguit el següent procediment.

Primerament, s'ha observat un paquet i dins del camp *Ethernet II* o *IEE 802.3 Ethernet* (en funció del paquet) s'ha observat la MAC del camp *Source*. Seguidament, s'ha afegit com a filtre els paquets IPX que no contenen aquella MAC. Per exemple, si s'observa l'adreça MAC aa:bb:cc:dd:ee:ff s'afegeix com a filtre de la següent manera:

```
ipx && eth.src != aa:bb:cc:dd:ee:ff
```

[9] HAL (*Hardware Address Length*): Mida, en bits, de l'adreça física.

[10] PAL (*Protocol Address Length*): Mida, en bits, de l'adreça del protocol de la capa superior.

[11] IPX (Internetwork Packet Exchange): Protocol utilitzat en xarxes *NetWare* de *Novell*.

Aquest procediment es fa de forma recursiva per a cada MAC diferent fins a que el filtre no mostri cap paquet. A continuació, a partir de la MAC es filtren tots els paquets que no corresponen a un paquet amb protocol IPX i que tinguin la MAC corresponent. Per exemple, per esbrinar l'adreça IPv4 d'un equip amb la MAC aa:bb:cc:dd:ee:ff s'aplica el següent filtre:

```
!ipx && eth.src == aa:bb:cc:dd:ee:ff
```

Mitjançant un paquet diferent a IPX es pot identificar l'adreça IP de l'equip dins del camp *Sender IP address* o *Source Address.* Seguint amb aquest procediment per a cada adreça MAC s'ha elaborat la següent taula:

Taula 10: Equips que empren IPX i IPv4

Adreça IPX	MAC	IPv4 associada
00000009.00080228befa	00:08:02:28:be:fa	172.16.30.7
00000000.000074e03eaf	00:00:74:e0:3e:af	172.16.40.5
00000000.00007499b50b	00:00:74:99:b5:0b	172.16.40.7
00000000.000074b4dbcd	00:00:74:b4:db:cd	172.16.0.81
00000009.001e0b120f90	00:1e:0b:12:0f:90	172.16.3.56

3.3. Adreces IPv4 associades a IPv6 per a MDNS

Per a mostrar els paquets que empren el protocol MDNS, s'aplica el següent filtre:

```
mdns
```

A partir de l'adreça IPv6 es pot obtenir la seva adreça MAC en la capçalera del camp *Ethernet II*. Amb aquesta adreça MAC es pot obtenir l'adreça IPv4 associada mitjançant el següent filtre:

```
!ipv6 && eth.addr == <MAC_Equip>
```

Com a resultat d'aquesta anàlisi, s'ha confeccionat la taula següent:

Taula 11: Adreces IPv4 que usen MDNS per a IPv6

IPv6	MAC	IPv4
fe80::21e:bff:fe12:f90	00:1e:0b:12:0f:90	172.16.3.56
fe80::1093:c39a:6dbe:4378	48:0f:cf:3e:5e:1a	172.16.12.6
fe80::4d5:32ca:ba89:5d43	b4:b5:2f:ba:8d:45	172.16.18.113
fe80::20f:feff:fe98:c253	00:0f:fe:98:c2:53	172.16.26.151
fe80::6d66:f968:cee6:5b9	00:0f:fe:7d:c4:ca	172.16.26.158
fe80::b6b5:2fff:feb2:41c	b4:b5:2f:b2:04:1c	172.16.51.20
fe80::26be:5ff:fe1c:edf7	24:be:05:1c:ed:f7	172.16.51.34
fe:80::b6b5:2fff:feb2:39f	b4:b5:2f:b2:03:9f	172.16.51.43
fe80::59e6:7baf:b603:1a4b	70:71:bc:5d:92:92	172.16.101.244

3.4. Adreces *multicast* IPv4

Per a mostrar únicament els paquets que empren adreces *multicast* en IPv4, es pot fer amb el següent filtre:

```
ip.addr >= 224.0.0.0 && ip.addr < 240.0.0.0 && !ipv6
```

Mitjançant aquest filtre es mostren únicament les adreces de la classe D, mostrada en la taula 1. En el camp *Destination* es poden observar totes les adreces *multicast* de IPv4. S'ha d'excloure també els paquets que empren el protocol DHCP, ja que empren una adreça de *broadcast*.

A continuació es mostra una taula amb les adreces *multicast* IPv4 presents en la captura:

Taula 12: Protocols de IPv4 emprats en adreces multicast

Adreça	Protocol
224.0.0.252	LLMNR
224.0.0.251	MDNS i IGMPv2
224.0.0.22	IGMPv3
224.0.0.18	VRRP
224.0.0.1	BJNP, IGMPv2

A continuació es proporciona una breu descripció per a cada protocol:

- **LLMNR** (*Link-Local Multicast Name Resolution*): És un protocol de resolució de noms que permet als dispositius d'una xarxa local comunicar-se entre si utilitzant noms de *host* sense dependre d'un servidor DNS.
- **MDNS** (*Multicast DNS*)**:** És un protocol que permet als dispositius d'una xarxa local resoldre adreces usant noms de *host* sense dependre d'un servidor DNS.
- **IGMPv2** (*Internet Group Management Protocol version 2*): És un protocol que facilita a un amfitrió indicar al seu encaminador local que vol participar en la transmissió d'un grup *multicast*.
- **IGMPv3** (*Internet Group Management Protocol version 3*): Aquest protocol millora el protocol IGMPv2 al permetre a un amfitrió especificar quines són les fonts de tràfic multicast que vol rebre, a més de la pertinença a un grup *multicast*.
- **VRRP** (*Virtual Router Redundancy Protocol*): És un protocol de redundància que permet diversos encaminadors en una xarxa compartir una única adreça IP.
- **BJNP** (*Canon's proprietary network printing protocol*): És un protocol utilitzat per les impressores *Canon* per a la impressió a través de la xarxa.

3.5. Gràfic dels protocols de nivell 3

Per a identificar els protocols de nivell 3 presents en la captura de la xarxa es pot fer mitjançant el menú *Estadísticas → Jerarquía de Protocolo*. Aquest menú mostra la jerarquia de protocols de la xarxa per a cada nivell. Dins del desplegable *Frame → Ethernet* es troben els protocols de nivell 3 de la xarxa. Mitjançant el botó de *Copiar como CSV* es poden obtenir aquestes dades.

Seguidament, aquestes dades s'han tractat i s'ha elaborat la següent taula amb els paquets segons cada protocol:

Taula 13: Protocols de nivell 3

Protocol	Nombre de paquets
ARP	21.303
IPv4	9.448
IPv6	2.657
LLC[12]	258
IPX	50

El nombre total de paquets és de 33716. Per a obtenir el nombre de paquets que usen el protocol IPv6 s'ha emprat el següent filtre per a descomptar els paquets IPv4:

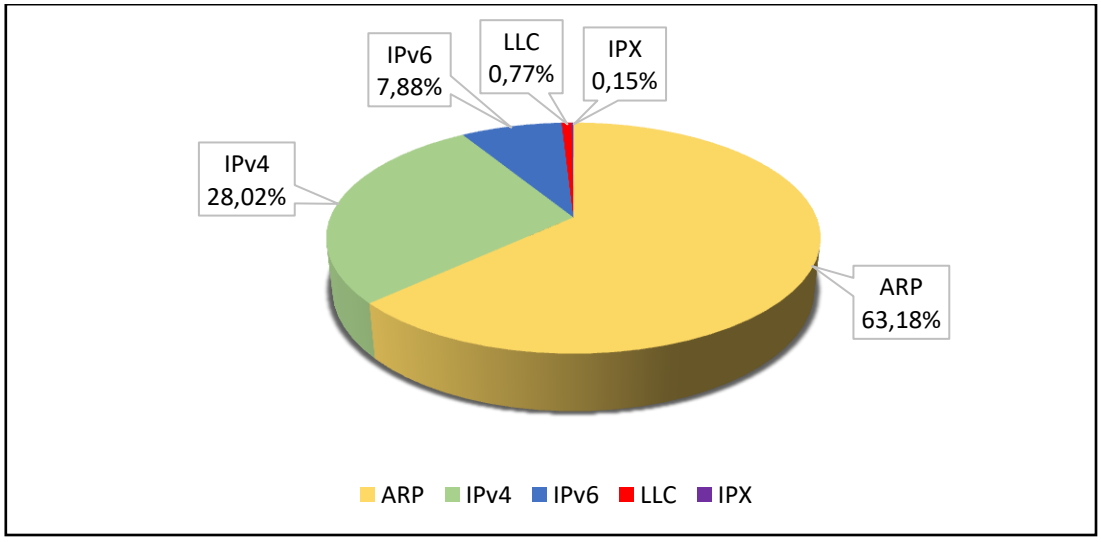
IPv6 → `!ip && ipv6`

Aquest filtre descompta els paquets *Teredo IPv6 over UDP tunneling*, ja que són paquets que empren un mecanisme de transició per a permetre la comunicació entre dispositius que empren IPv6 i IPv4.

[12] LCC (Logical-Link Control): Control d'enllaç de dades en xarxes com *Ethernet*.

Des del punt de vista de nivell 3, aquests paquets són IPv4 i, per tant, s'han de descomptar d'IPv6. A partir de l'anterior taula s'ha elaborat un gràfic de pastís dels diferents protocols de nivell 3 presents en la captura de xarxa:

Il·lustració 1: Gràfic dels protocols de nivell 3



4. Anàlisi nivell de transport

L'anàlisi a nivell de transport implica l'estudi dels protocols de transport com TCP i UDP. També, s'analitzen els mecanismes relacionats a aquests com el control de congestió, el control d'errors, i el control de flux de dades. El principal objectiu és avaluar el rendiment dels protocols de transport i detectar errors en les comunicacions presents en la captura.

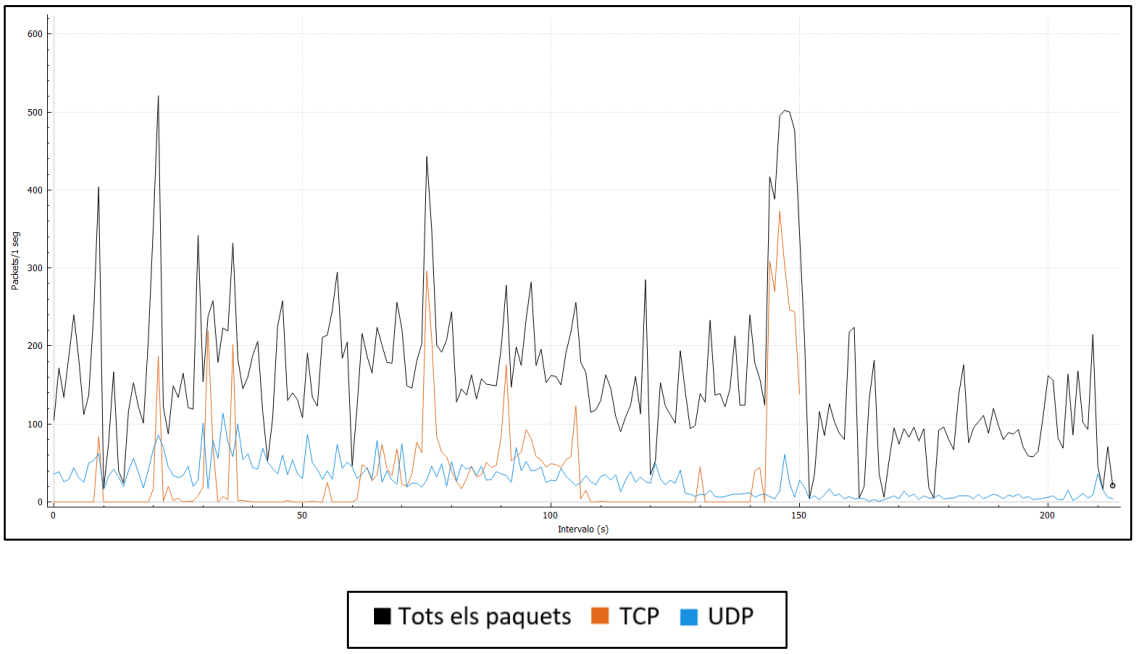
Per a realitzar aquest anàlisi, a partir de la secció 4.1. s'han desestimat els paquets que tenen com a adreça destí l'adreça de *broadcast* de nivell 2. També s'han desestimat els paquets *multicast*, IPv6 i els protocols ARP, DNS i NTP. Per a evitar mostrar aquests paquets s'ha afegit el següent filtre a tots els filtres emprats:

```
eth.dst != ff:ff:ff:ff:ff:ff && (!ip.addr >= 224.0.0.0 ||
  !ip.addr < 240.0.0.0) && !ipv6 && !arp && !dns && !ntp
```

4.1. Gràfica de trànsit en el temps de TCP/IP

A continuació es mostra un gràfic del trànsit de la captura en funció del temps. Aquest gràfic està mesurat en paquets en intervals d'un segon. En aquest gràfic es mostra el trànsit total de la captura, i els dos protocols de transport de TCP/IP, és a dir, TCP i UDP.

Il·lustració 2: Gràfic del trànsit en el temps de TCP/IP



Per a mostrar els paquets TCP i UDP s'ha afegit una nova fila on s'han aplicat els filtres `tcp` i `udp` respectivament.

4.2. Comunicacions TCP que no s'han dut a terme

Per a mostrar les comunicacions TCP que no s'han dut a terme és adequat identificar paquets amb el *flag* SYN enviat però sense cap paquet de resposta SYN, ACK (SYN !ACK). Aquests paquets representen els intents de comunicació que no s'han completat.

També és adequat identificar paquets amb el *flag* RST (*Reset*) abans d'un paquet amb el *flag* FIN, el qual indica que s'ha produït un error en la comunicació.

Per a visualitzar els paquets amb el *flag* SYN present, però sense el *flag* ACK, s'ha emprat el següent filtre:

```
tcp.flags.syn == 1 && tcp.flags.ack == 0
```

Per a cada comunicació mostrada per aquest filtre mitjançant el menú contextual de *Filtro de Conversacion → TCP* es pot veure la conversa sencera. A partir de la conversa es pot determinar si hi ha hagut comunicació o no en funció de si s'ha rebut un ACK del paquet enviat. D'altra banda, per a mostrar els paquets amb el *flag* RST present, s'ha aplicat el següent filtre:

```
tcp.flags.reset == 1
```

A partir dels paquets observats, s'ha elaborat la següent taula:

Taula 14: Comunicacions TCP no dutes a terme

IP-Origen	Port-Origen	IP-Destí	Port-Destí	Motiu-Fallida
172.16.0.102	43384	172.16.0.111	1759	SYN !ACK
172.16.0.105	51213	10.50.54.87	1243	SYN !ACK
172.16.0.108	44417	172.16.0.103	7111	RST
172.16.0.109	55136	75.52.4.122	10576	SYN !ACK
172.16.0.109	60523	172.16.0.106	5391	RST
172.16.0.110	32980	172.16.0.105	11243	RST
172.16.0.111	43660	172.16.0.114	9118	RST

4.3. Comunicacions TCP completes

Una comunicació TCP és considerada complerta si es produeix tant el procediment d'inici de comunicació com el de finalització sense una pèrdua de connexió. Per a determinar les comunicacions TCP que s'han completat correctament, s'ha separat en dos tipus principalment.

4.3.1. Comunicacions HTTP i HTTPS

En aquestes connexions no es consideren les connexions realitzades per a cada element de la pàgina. En el seu lloc, es considera coma una única comunicació les connexions des d'una mateixa adreça origen cap a la mateixa adreça destí i port 80 o 443. Per a filtrar els paquets amb aquests requisits s'ha afegit el següent filtre:

```
tcp.port == 80 || tcp.port == 443
```

Aquest procediment s'ha efectuat per els paquets del tipus HTTP i HTTPS de la mateixa forma que a l'apartat anterior. A partir de les dades observades s'han confeccionat les següents taules:

Taula 15: Comunicacions HTTP completes

IP-Origen	IP-Destí
172.16.0.102	152.19.134.142
172.16.0.103	85.236.55.6
172.16.0.103	209.132.181.16
172.16.0.109	84.88.27.7
172.16.0.110	91.216.63.240
172.16.0.110	130.206.192.24
172.16.0.115	216.58.210.131
172.16.0.117	95.101.112.50

En el cas dels paquets que empren el protocol d'HTTPS no s'ha observat cap comunicació exitosa, ja que l'única comunicació present no finalitza correctament la comunicació.

4.3.2. Resta de comunicacions TCP

D'altra banda, s'han analitzat els paquets que empren el protocol TCP que no son ni HTTP ni HTTPS. Per a evitar aquests paquets, s'ha negat el filtre del apartat anterior. A més, també s'han exclòs els paquets TCP que no tenen el *flag* de FIN, ja que aquests no formen part de les comunicacions TCP completes:

```
tcp.port != 80 && tcp.port != 443 && tcp.flags.fin == 1
```

Per a calcular la finestra origen inicial (FOI) es realitza a mitjançant el paquet amb el *flag* SYN i fer l'operació de *window_size * window_scale*. Aquests camps es poden observar dins de TCP, i en el cas de *window_scale*, dins del subcamp *options*.

La mateixa operació s'ha realitzat per a finestra destí inicial (FDI) però amb els paquets que contenen el *flag* SYN,ACK. D'altra banda, la MTU sempre és 1500 bytes per defecte en TCP.

La descripció de les sigles emprades per a aquesta taula és la següent:

- **PO**: Port-Origen
- **MO**: MTU-Origen
- **FOI**: Finestra-Origen-Inicial
- **PD**: Port-Destí
- **MD**: MTU-Destí
- **FDI**: Finestra-Destí-Inicial

Taula 16: Resta de comunicacions TCP

IP-Origen	PO	MO	FOI	IP-Destí	PD	MD	FDI
172.16.0.102	46148	1500	3737600	172.16.0.106	21	1520	3706880
172.16.0.102	34663	1500	3737600	172.16.0.110	2483	1500	3706880
172.16.0.107	43457	1500	3737600	172.16.0.112	4931	1500	3706880
172.16.0.110	38085	1500	3737600	172.16.0.105	32458	1500	3706880
172.16.0.113	48213	1500	3737600	172.16.0.110	4643	1500	3706880
172.16.0.117	53120	1500	3737600	172.16.0.104	22	1500	3706880
172.16.0.117	54868	1500	3737600	172.16.0.124	8164	1500	3706880

4.4. Comunicacions UDP que no s'han dut a terme

Primerament, per a mostrar les comunicacions que empren únicament UDP es pot fer mitjançant el menú *Estadísticas → Conversaciones*. És adequat considerar que aquest procediment s'ha d'emprar aplicant el filtre especificat a l'apartat 4. Seguidament, seleccionar la casella del protocol UDP per a veure totes les converses que empren purament aquest protocol.

Per a garantir mostrar únicament aquelles conversacions que no s'han dut a terme, és adequat verificar si és present un paquet ICMP de caràcter *Port Unreachable* o *Host Unreachable*. Per a verificar això, s'aplica el següent filtre amb les adreces de les comunicacions i un port:

```
udp && ip.src == <IP-Origen> && ip.dst == <IP-Destí> &&
        udp.port == <Port-Origen_o_Port-Destí>
```

Taula 17: Comunicacions UDP no dutes a terme

IP-Origen	Port-Origen	IP-Destí	Port-Destí	Motiu-Fallida
172.16.0.105	50062	172.16.0.114	7556	*Port unreachable*
172.16.0.108	58486	172.16.0.110	6914	*Port unreachable*
172.16.0.116	37758	172.16.0.119	11345	*Port unreachable*
172.16.0.117	39679	172.16.0.121	8241	*Port unreachable*

També s'han detectat una sèrie d'equips que empren *traceroute*. Aquesta és una eina utilitzada per a identificar la ruta que prenen els paquets de dades des del seu origen fins al seu destí a través de la xarxa. El filtre que permet capturar aquests paquets és el següent

```
udp.port >= 33434 && udp.port <= 33534
```

Tanmateix, és adequat considerar que aquest filtre també pot capturar altres paquets UDP que empren aquests ports, però s'han d'observar aquells paquets ICMP que el *Time-To-Live* (*TTL*) ha estat superat. D'aquesta manera, s'ha elaborat la següent taula:

Taula 18: Dispositius que empren Traceroute

IP-Origen	Port-Origen	IP-Destí	Port-Destí
172.16.0.105	50062	172.16.0.114	7556
172.16.0.115	56308	172.16.116	27823

4.5. Comunicacions UDP que s'han dut a terme

Per a mostrar els paquets que empren purament el protocol UDP s'ha emprat la mateixa metodologia que a l'apartat anterior. En aquest cas mitjançant el filtre de conversa s'ha observat les comunicacions que s'han dut a terme. Per a garantir-ho, s'aplica el següent filtre per tal de mostrar si hi ha hagut un paquet ICMP que mostri algun error:

```
udp && ip.src == <IP-Origen> && ip.dst == <IP-Destí> &&
        udp.port == <Port-Origen_o_Port-Destí>
```

En el cas que es mostri un paquet ICMP del tipus *Time-to-live exceeded* o *Port unreachable*, no es pot garantir que el paquet s'ha dut a terme. D'altra banda, les comunicacions que s'han detectat com a *traceroute* no han estat incloses, ja que no es pot garantir que la comunicació s'ha dut a terme. Mitjançant aquest procediment, s'ha elaborat la següent taula amb els paquets que sí que es pot garantir que s'han dut a terme.

Taula 19: Comunicacions UDP dutes a terme

IP-Origen	Port-Origen	IP-Destí	Port-Destí
172.16.0.102	51989	172.16.0.105	6396
172.16.0.104	36582	80.47.30.2	6591
172.16.0.107	60907	172.16.0.109	18599
172.16.0.108	58647	172.16.0.104	8937
172.16.0.110	44749	10.0.0.1	8131
172.16.0.115	56308	172.16.116	27823
172.16.0.116	55976	172.16.0.119	5822

4.6. Les tres comunicacions TCP amb més transferència de dades.

Primerament, per a identificar les tres comunicacions TCP que duen una major transferència de dades, s'ha mostrat mitjançant el menú de *Estadísticas → Conversaciones*. En aquest menú es poden veure les diferents comunicacions TCP, les quals es poden ordenar per quantitat de *bytes* i comparar-les amb les obtingudes en l'apartat 4.3.2.

Les tres comunicacions que s'han emprat per a determinar les obtingudes han estat les següents:

Taula 20: Comunicacions TCP amb més transferència de dades

IP-Origen	Port-Origen	IP-Destí	Port-Destí
172.16.0.107	43457	172.16.0.112	4931
172.16.0.110	38085	172.16.0.105	32458
172.16.0.117	54868	172.16.0.124	8164

4.6.1. Paquets d'apertura i tancament de connexió

A continuació es mostra una taula per a cada comunicació amb el número assignat per Wireshark en la captura d'inici i de tancament de la connexió. Per a realitzar això s'ha aplicat la següent comanda per a cada comunicació:

```
ip.addr == <IP-Origen> && ip.addr == <IP-Destí>
```

S'han observat cada tipus de paquet d'apertura i tancament per a obtenir el número assignat al paquet. A partir de l'anàlisi realitzada, s'han confeccionat les següents taules:

Taula 21: Paquets d'apertura i tancament entre 172.16.0.107 i 172.16.0.112

Paquets d'apertura	Número Inici	Paquets de tancament	Número Tancament
SYN	19208	FIN, PSH, ACK	19333
SYN, ACK	19209	FIN, ACK	19421
ACK	19210	ACK	19422

Taula 22: Paquets d'apertura i tancament entre 172.16.0.110 i 172.16.0.105

Paquets d'apertura	Número Inici	Paquets de tancament	Número Tancament
SYN	1520	FIN, PSH, ACK	1834
SYN, ACK	1521	FIN, ACK	1845
ACK	1522	ACK	1847

Taula 23: Paquets d'apertura i tancament entre 172.16.0.117 i 172.16.0.124

Paquets d'apertura	Número Inici	Paquets de tancament	Número Tancament
SYN	9488	FIN, PSH, ACK	9532
SYN, ACK	9489	FIN, ACK	9533
ACK	9490	ACK	9534

4.6.2. Nombre de bytes d'usuari i total transmesos en cada sentit de la comunicació

Per a veure el nombre de bytes emprats per a la comunicació, primerament s'ha aplicat el següent filtre per a cada una de les tres comunicacions:

```
ip.addr == <IP-Origen> && ip.addr == <IP-Destí>
```

A l'aplicar aquest filtre primerament s'ha identificat els *Bytes* totals emprats en la comunicació en el menú *Estadísticas → Conversaciones → TCP*. Els *Bytes* totals és el conjunt de *Bytes* emprats en la comunicació, aquests impliquen tant els *Bytes* de dades com els de capçaleres de cada protocol. En les columnes *Bytes A → Bytes B* i *Bytes B → Bytes A*, es pot observar la quantitat total de bytes enviats en cada direcció de la comunicació.

D'altra banda, per a obtenir el nombre de bytes d'aplicació s'ha d'observar mitjançant el menú contextual de *Seguir → Secuencia TCP* al seleccionar un paquet. Aquest menú proporciona en kB el total de dades d'usuari per a cada sentit de la comunicació. En el cas de la comunicació de B → A les dades són 0 kB, ja que només hi ha paquets ACK.

És important recordar que les dades dels *Bytes* totals s'han obtingut en *Kibibytes* (KiB), així que s'ha de realitzar una conversió. Cal recordar que 1 KiB és igual a 1,024 k*Bytes*. Emprant aquesta metodologia s'han elaborat les següents taules:

Taula 24: KiloBytes totals per a cada sentit de les comunicacions

Adreça IP A	**Adreça IP B**	***Total Bytes A → B***	***Total Bytes B → A***
172.16.0.107	172.16.0.112	89,434 kB	4,200 kB
172.16.0.110	172.16.0.105	64,244 kB	2,908 kB
172.16.0.117	172.16.0.124	16,438 kB	0,654 kB

Taula 25: KiloBytes d'usuari per a cada sentit de les comunicacions

Adreça IP A	**Adreça IP B**	***Bytes* usuari *A → B***	***Bytes usuari B → A***
172.16.0.107	172.16.0.112	19 kB	0 kB
172.16.0.110	172.16.0.105	19 kB	0 kB
172.16.0.117	172.16.0.124	9,648 kB	0 kB

4.6.3. Opcions TCP intercanviades en la fase de connexió

La fase de connexió TCP consta de tres paquets ([SYN], [SYN, ACK], [ACK]). En cada un d'aquests paquets es poden observar les diferents opcions TCP (Menú *Options*). S'ha desplegat cada opció i s'ha emprat el menú contextual *Copiar → Todos los ítems del árbol seleccionados*.

Comunicació 172.16.0.107 ⇔ 172.16.0.112:

[SYN]:

```
Options: (20 bytes), Maximum segment size, SACK permitted,
Timestamps, No-Operation (NOP), Window scale
    TCP Option - Maximum segment size: 1460 bytes
        Kind: Maximum Segment Size (2)
        Length: 4
        MSS Value: 1460
    TCP Option - SACK permitted
        Kind: SACK Permitted (4)
        Length: 2
    TCP Option - Timestamps
        Kind: Time Stamp Option (8)
        Length: 10
        Timestamp value: 1191718: TSval 1191718, TSecr 0
        Timestamp echo reply: 0
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - Window scale: 7 (multiply by 128)
        Kind: Window Scale (3)
        Length: 3
        Shift count: 7
        [Multiplier: 128]
```

[SYN, ACK]:

```
Options: (20 bytes), Maximum segment size, SACK permitted,
Timestamps, No-Operation (NOP), Window scale
    TCP Option - Maximum segment size: 1460 bytes
        Kind: Maximum Segment Size (2)
        Length: 4
        MSS Value: 1460
    TCP Option - SACK permitted
        Kind: SACK Permitted (4)
        Length: 2
    TCP Option - Timestamps
        Kind: Time Stamp Option (8)
        Length: 10
        Timestamp value: 1203724: TSval 1203724, TSecr
1191718
        Timestamp echo reply: 1191718
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - Window scale: 7 (multiply by 128)
        Kind: Window Scale (3)
        Length: 3
        Shift count: 7
        [Multiplier: 128]
```

[ACK]:

```
Options: (12 bytes), No-Operation (NOP), No-Operation
(NOP), Timestamps
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - Timestamps
        Kind: Time Stamp Option (8)
        Length: 10
```

```
        Timestamp value: 1191719: TSval 1191719, TSecr
1203724
        Timestamp echo reply: 1203724
```

Comunicació 172.16.0.110 ⇔ 172.16.0.105:

[SYN]:

```
Options: (20 bytes), Maximum segment size, SACK permitted,
Timestamps, No-Operation (NOP), Window scale
    TCP Option - Maximum segment size: 1460 bytes
        Kind: Maximum Segment Size (2)
        Length: 4
        MSS Value: 1460
    TCP Option - SACK permitted
        Kind: SACK Permitted (4)
        Length: 2
    TCP Option - Timestamps
        Kind: Time Stamp Option (8)
        Length: 10
        Timestamp value: 4935522: TSval 4935522, TSecr 0
        Timestamp echo reply: 0
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - Window scale: 7 (multiply by 128)
        Kind: Window Scale (3)
        Length: 3
        Shift count: 7
        [Multiplier: 128]
```

[SYN, ACK]:

```
Options: (20 bytes), Maximum segment size, SACK permitted,
Timestamps, No-Operation (NOP), Window scale
    TCP Option - Maximum segment size: 1460 bytes
        Kind: Maximum Segment Size (2)
        Length: 4
```

```
        MSS Value: 1460
    TCP Option - SACK permitted
        Kind: SACK Permitted (4)
        Length: 2
    TCP Option - Timestamps
        Kind: Time Stamp Option (8)
        Length: 10
        Timestamp value: 4939515: TSval 4939515, TSecr
4935522
        Timestamp echo reply: 4935522
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - Window scale: 7 (multiply by 128)
        Kind: Window Scale (3)
        Length: 3
        Shift count: 7
        [Multiplier: 128]
```

[ACK]:

```
Options: (12 bytes), No-Operation (NOP), No-Operation
(NOP), Timestamps
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - Timestamps
        Kind: Time Stamp Option (8)
        Length: 10
        Timestamp value: 4935523: TSval 4935523, TSecr
4939515
        Timestamp echo reply: 4939515
```

Comunicació 172.16.0.117 ⇔ 172.16.0.124:

[SYN]:

```
Options: (20 bytes), Maximum segment size, SACK permitted,
Timestamps, No-Operation (NOP), Window scale
    TCP Option - Maximum segment size: 1460 bytes
        Kind: Maximum Segment Size (2)
        Length: 4
        MSS Value: 1460
    TCP Option - SACK permitted
        Kind: SACK Permitted (4)
        Length: 2
    TCP Option - Timestamps
        Kind: Time Stamp Option (8)
        Length: 10
        Timestamp value: 3223872613: TSval 3223872613,
TSecr 0
        Timestamp echo reply: 0
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - Window scale: 7 (multiply by 128)
        Kind: Window Scale (3)
        Length: 3
        Shift count: 7
        [Multiplier: 128]
```

[SYN, ACK]:

```
Options: (20 bytes), Maximum segment size, SACK permitted,
Timestamps, No-Operation (NOP), Window scale
    TCP Option - Maximum segment size: 1460 bytes
        Kind: Maximum Segment Size (2)
        Length: 4
        MSS Value: 1460
    TCP Option - SACK permitted
        Kind: SACK Permitted (4)
```

```
        Length: 2
    TCP Option - Timestamps
        Kind: Time Stamp Option (8)
        Length: 10
        Timestamp value: 5067619: TSval 5067619, TSecr
3223872613
        Timestamp echo reply: 3223872613
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - Window scale: 7 (multiply by 128)
        Kind: Window Scale (3)
        Length: 3
        Shift count: 7
        [Multiplier: 128]
```

[ACK]:

```
Options: (12 bytes), No-Operation (NOP), No-Operation
(NOP), Timestamps
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - No-Operation (NOP)
        Kind: No-Operation (1)
    TCP Option - Timestamps
        Kind: Time Stamp Option (8)
        Length: 10
        Timestamp value: 3223872614: TSval 3223872614,
TSecr 5067619
        Timestamp echo reply: 5067619
```

En aquest context, és adequat proporcionar una descripció de la utilitat de cada opció TCP.

- ***TCP Option - Maximum segment size***: És una opció que permet especificar la mida màxima dels segments de dades que es

poden enviar en una única trama de la capa de xarxa. Aquesta opció s'especifica en el procés d'inici de comunicació TCP.

- ***TCP Option - SACK permitted***: És una opció que permet al receptor confirmar la recepció de paquets fora d'orde en la trama. Aquesta opció permet una recuperació més eficient dels paquets perduts. D'igual manera, aquesta opció només està present en el procés d'inici de la comunicació.
- ***TCP Option - Timestamps***: És una opció que permet a cada paquet TCP portar una marca de temps. Aquesta marca és utilitzada per a diverses finalitats, principalment per calcular el temps de trànsit dels paquets (*Round-Trip Time*).
- ***TCP Option - No-Operation (NOP)***: És una opció que no té cap utilitat directa en la comunicació, únicament actua com a farciment per les altres opcions TCP.
- ***TCP Option - Window scale***: És una opció que permet a TCP emprar una mida de finestra major a l'original (64 KiB). Aquesta opció s'empra únicament en el procés d'inici de la comunicació.

4.6.4. Nombre de seqüencia inicial real del transmissor i receptor

Per a identificar el nombre de seqüència inicial real de la comunicació es pot veure en el camp de *Sequence Number* dins de TCP. En el cas del transmissor en el primer paquet de la seqüència d'inici ([SYN]). En el cas del receptor, en el segon paquet de la seqüència d'inici ([SYN, ACK]).

Taula 26: Nombre de seqüencia inicial real per a cada comunicació

Comunicació	Seq. real del transmissor	Seq. real del receptor
172.16.0.107 ⇔ 172.16.0.112	1691116532	1470491335
172.16.0.110 ⇔ 172.16.0.105	3859645030	2727988100
172.16.0.117 ⇔ 172.16.0.124	1064479517	829581103

5. Conclusions

Aquesta pràctica ha permès realitzar una anàlisi detallada del trànsit d'una xarxa utilitzant l'eina Wireshark. S'ha pogut observar la gran quantitat d'informació que es pot extreure d'una captura de xarxa, des de les adreces IP i MAC dels dispositius fins als protocols de comunicació que utilitzen.

S'ha identificat que la xarxa analitzada és de classe B, amb una adreça de xarxa de 172.16.0.0/16. L'adreça de *broadcast* és 172.16.255.255, l'encaminador té l'adreça IP 172.16.20.1 i l'adreça IP del servidor DNS és la 172.16.20.10.

En relació als protocols de trames de nivell 2, s'han identificat *Ethernet II* i *IEEE 802.3 Ethernet*. Dins d'aquests, s'han observat diversos protocols de nivell 3 encapsulats, com ara IPv4, IPv6, ARP i IPX/SPX.

S'ha observat que alguns equips utilitzen tant el protocol IPX com IPv4, i s'han identificat les seves adreces IP associades. També s'han identificat les adreces IPv4 que utilitzen MDNS per a IPv6.

En l'anàlisi del nivell de transport, s'han observat comunicacions TCP i UDP, algunes de les quals no s'han dut a terme. També s'ha realitzat una anàlisi més a detall a partir de les tres comunicacions TCP amb més transferència de dades.

En resum, aquesta pràctica ha proporcionat una visió global del programari Wireshark. La realització d'aquesta pràctica ha permès adquirir un coneixement general d'aquesta eina i de la seva utilitat en el context de xarxes.