



UNIVERSITAT<sup>DE</sup>  
BARCELONA

---

## Pràctica 5. El model OSI

---

Antoni Tuduri & Alejandro Guzman

8 gener 2023

## ÍNDICE

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Objectius de la pràctica</b>            | <b>3</b>  |
| <b>2</b> | <b>Exercici 1</b>                          | <b>3</b>  |
| 1        | Obtenció de les adreces IP i MAC . . . . . | 3         |
| 2        | Filtratge per IP . . . . .                 | 3         |
| 3        | Estructura de la direcció MAC . . . . .    | 4         |
| 4        | Diferents camps capçalera IP . . . . .     | 5         |
| <b>3</b> | <b>Exercici 2</b>                          | <b>5</b>  |
| 1        | Diagrama temporal de la connexió . . . . . | 6         |
| <b>4</b> | <b>Exercici 3</b>                          | <b>7</b>  |
| <b>5</b> | <b>Exercici 4</b>                          | <b>9</b>  |
| <b>6</b> | <b>Conclusions</b>                         | <b>12</b> |

## 1. OBJECTIUS DE LA PRÀCTICA

L'objectiu principal de la pràctica és veure com s'encapsulen les diferents Unitats de Protocol d'Usuari (DPU) i que permeten transmetre informació entre dos equips de manera estàndard, independentment de les característiques dels equips en qüestió.

## 2. EXERCICI 1

El primer pas serà monitorar tot el trànsit de la nostra xarxa local i filtrar-la mostrant únicament els paquets que impliquin directament la nostra IP. D'aquests paquets obtindrem posteriorment la seva informació relativa a les adreces MAC i IP, desglossant-la per poder explicar quina és la funció de cadascun dels bits.

### 1. Obtenció de les adreces IP i MAC

Les adreces IP i MAC les esbrinarem executant la comanda `ipconfig/all` en la consola de Windows.

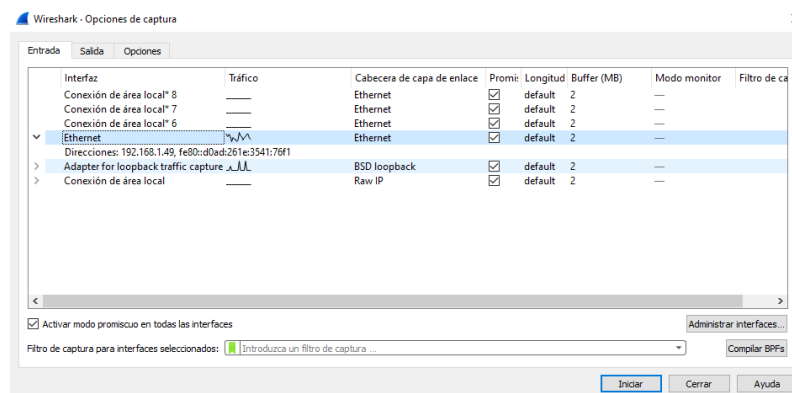
```
Adaptador de Ethernet Ethernet:
    Sufijo DNS específico para la conexión. . . :
    Descripción. . . . . : Realtek Gaming GbE Family Controller
    Dirección física. . . . . : B4-2E-99-51-75-C0
    DHCP habilitado. . . . . : sí
    Configuración automática habilitada. . . . : sí
    Vínculo: dirección IPv6 local. . . : fe80::d0ad:261e:3541:76f1%6(Preferido)
    Dirección IPv4. . . . . : 192.168.1.49(Preferido)
    Máscara de subred. . . . . : 255.255.255.0
    Concesión obtenida. . . . . : jueves, 5 de enero de 2023 16:12:38
    La concesión expira. . . . . : viernes, 6 de enero de 2023 4:12:35
    Puerta de enlace predeterminada. . . . . : 192.168.1.1
    Servidor DHCP. . . . . : 192.168.1.1
```

Figura 2.1: Comanda `ipconfig /all`

Com es pot comprovar, la direcció IP associada a aquesta interfície de xarxa és 192.168.1.49 i la direcció MAC és B4-2E-99-51-75-C0.

### 2. Filtratge per IP

Havent obtingut el valor de la nostra adreça IP ara podem procedir a filtrar el trànsit de la nostra xarxa local mostrant únicament aquells paquets en els quals se segueixi el protocol TCP i es vegi implicada la nostra adreça IP, ja sigui com a adreça origen o adreça destí.



I ara aplicarem un filtre a aquestes dades fent que només es mostrin aquelles en les quals figurei la nostra IP.

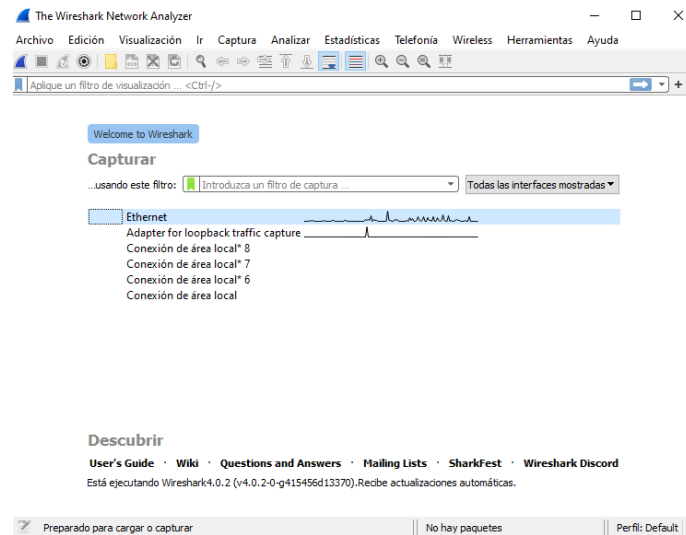


Figura 2.2: Cliquem dins la interfície Ethernet

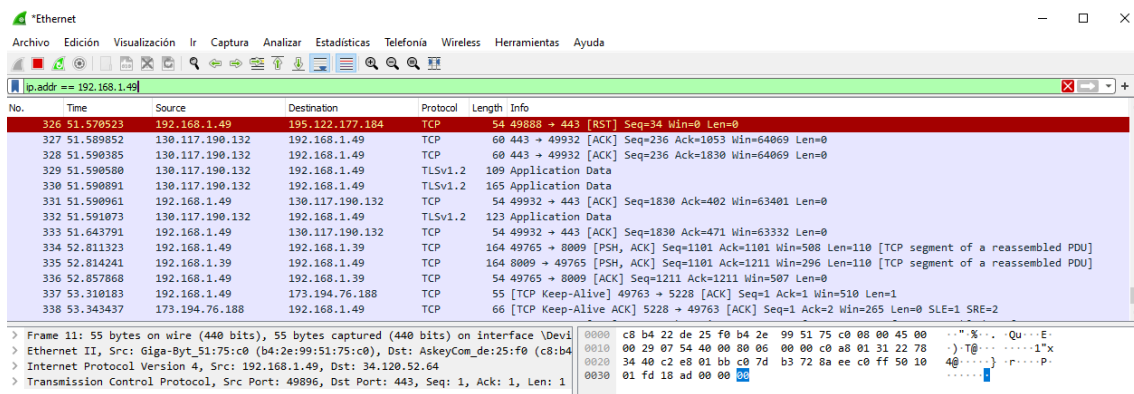


Figura 2.3: Filtrem per la nostra IP

### 3. Estructura de la direcció MAC

Un cop hem seleccionat un paquet amb el protocol TCP podem veure al apartat Ethernet II dades sobre la nostra direcció MAC i sobre el destinatari. Allà podem comprovar que la direcció MAC té dos bits remarcats, les funcions de les quals són les següents:

**LG Bit:** Indica si la direcció MAC ha estat assignada per un proveïdor o administrativament.

**IG Bit:** Indica si la direcció MAC és individual o de grup, valent 0 si és d'unidifusió o valent 1 si és de multidifusió o broadcast.

```

▼ Ethernet II, Src: Giga-Byt_51:75:c0 (b4:2e:99:51:75:c0), Dst: Google_54:3c:96 (44:07:0b:54:3c:96)
  ▼ Destination: Google_54:3c:96 (44:07:0b:54:3c:96)
    Address: Google_54:3c:96 (44:07:0b:54:3c:96)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  ▼ Source: Giga-Byt_51:75:c0 (b4:2e:99:51:75:c0)
    Address: Giga-Byt_51:75:c0 (b4:2e:99:51:75:c0)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
  > Internet Protocol Version 4, Src: 192.168.1.49, Dst: 192.168.1.39
  > Transmission Control Protocol, Src Port: 49765, Dst Port: 8009, Seq: 221, Ack: 221, Len: 0

```

Figura 2.4: Ethernet II

#### 4. Diferents camps capçalera IP

Un altre cop el programa ens ofereix informació de com està formada la nostra estructura de la direcció IP.

```

▼ Internet Protocol Version 4, Src: 192.168.1.49, Dst: 192.168.1.39
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x8b2f (35631)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: TCP (6)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.49
    Destination Address: 192.168.1.39

```

Figura 2.5: Camps capçalera IP

**Version:** Ens indica si segueix el protocol IPv4 o IPv6, aquest cas com ens diu en nombre binari 01002 es la v4.

**Header length:** Longitud de la capçalera.

**Differentiated Services Field :** Informació de la qualitat del servei de comunicació

-**Differentiated Services Codepoint:**Diferenciar la qualitat de transmissió que tenen les dades que es transmeten.

-**Explicit Congestion Notification:** Notificació de la congestió entre endpoints.

**Total length:** Mida total del datagrama.

**Identification:**Identificador únic del datagrama.

**Flags:** Indicar si el paquet es pot fragmentar o no.

– **Reserved Bit:** Sempre 0

– **Don't fragment**

– **More fragments**

**Fragment offset:** Posició del fragment dins del paquet.

**Time to live:** Nombre de nodes pels quals pot ser enviat abans de ser descartat.

**Protocol**

**Header checksum :** detecció d'errors.

**Source address**

**Destination address**

### 3. EXERCICI 2

Procedim a capturar les comunicacions llençades per: telnet time-A.timefreq.bldrdoc.gov 13

```

59949 23-01-05 16:19:36 00 0 0 0.0 UTC(NIST) *
Se ha perdido la conexión con el host.

```

Figura 3.1: Comanda telnet time-A.timefreq.bldrdoc.gov 13

El que hem realitzat amb l'anterior comanda ha sigut connectar-nos a un servidor remot. Si volem averiguar quina és l'adreça IP d'aquesta màquina haurem de buscar a Wireshark un pa-

quet amb origen a la nostra adreça IP (192.168.1.49) i que utilitzi el protocol DNS.

| No. | Time     | Source       | Destination     | Protocol | Length | Info   |
|-----|----------|--------------|-----------------|----------|--------|--|
| 1   | 0.000000 | 192.168.1.49 | 192.168.1.39    | TCP      | 164    | 49775 → 8089 [PSH, ACK] Seq=1 Win=588 Len=110 [TCP segment of a reassembled PDU]                     |
| 2   | 0.002870 | 192.168.1.39 | 192.168.1.49    | TCP      | 164    | 8089 → 49775 [PSH, ACK] Seq=1 Win=279 Len=110 [TCP segment of a reassembled PDU]                     |
| 3   | 0.004551 | 192.168.1.49 | 192.168.1.39    | TCP      | 54     | 49775 → 8089 [ACK] Seq=111 Win=507 Len=0   |
| 4   | 0.405759 | 34.120.52.64 | 192.168.1.49    | TLSPV1.2 | 81     | Application Data   |
| 5   | 0.408443 | 192.168.1.49 | 34.120.52.64    | TLSPV1.2 | 85     | Application Data   |
| 6   | 0.508130 | 34.120.52.64 | 192.168.1.49    | TCP      | 60     | 443 → 49814 [ACK] Seq=28 Ack=32 Win=272 Len=0  |
| 21  | 2.516357 | 192.168.1.49 | 224.0.0.251     | IGMPv2   | 46     | Membership Report group 224.0.0.251  |
| 22  | 2.516509 | 192.168.1.49 | 224.0.0.252     | IGMPv2   | 46     | Membership Report group 224.0.0.252  |
| 23  | 2.516623 | 192.168.1.49 | 239.255.255.250 | IGMPv2   | 46     | Membership Report group 239.255.255.250  |
| 24  | 2.516732 | 192.168.1.49 | 239.255.255.253 | IGMPv2   | 46     | Membership Report group 239.255.255.253  |
| 32  | 3.963853 | 192.168.1.49 | 80.58.61.250    | DNS      | 87     | Standard query query 0xaaff9 A time-a.timefreq.bldrdoc.gov   |
| 33  | 3.976148 | 80.58.61.250 | 192.168.1.49    | DNS      | 131    | Standard query response 0xaaff9 A time-a.timefreq.bldrdoc.gov CNAME time-a-b.nist.gov A 132.163.96.1 |

Un cop trobat el paquet de dades podem veure la seva informació. Ens diu que ens hem connectat al port 53 i hem utilitzat el protocol UDP.

Aquest protocol DNS utilitza d'altres protocols com UDP i TCP com a protocol de transport i establir connexió amb el servidor DNS, respectivament.

User Datagram Protocol, Src Port: 63989, Dst Port: 53

Source Port: 63989  
 Destination Port: 53  
 Length: 53  
 Checksum: 0x5054 [unverified]  
 [Checksum Status: Unverified]  
 [Stream index: 3]  
 > [Timestamps]  
 UDP payload (45 bytes)

El diagrama de seqüència de l'intercanvi de control DNS és el següent:

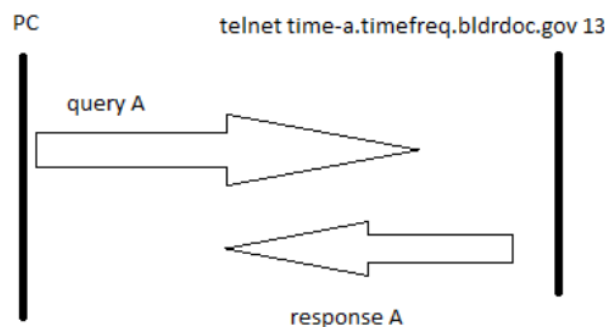


Figura 3.2: Diagrama seqüència intercanvi de control DNS

## 1. Diagrama temporal de la connexió

Primerament es realitzarà un intercanvi de comunicacions entre la nostra màquina i el servidor DNS per tal de conèixer quina és l'adreça IP d'aquest últim. Aquest intercanvi d'informació Una vegada coneguda l'adreça IP destí, proporcionada pel servidor DNS, es pot apreciar el següent intercanvi de control, que es pot veure a Wireshark. Aquest intercanvi de control es produeix a alt nivell de TCP.

|    |          |              |              |        |     |  |
|----|----------|--------------|--------------|--------|-----|--|
| 32 | 3.963853 | 192.168.1.49 | 80.58.61.250 | DNS    | 87  | Standard query 0xaaff9 A time-a.timefreq.bldrdoc.gov   |
| 33 | 3.976148 | 80.58.61.250 | 192.168.1.49 | DNS    | 131 | Standard query response 0xaaff9 A time-a.timefreq.bldrdoc.gov CNAME time-a-b.nist.gov A 132.163.96.1 |
| 34 | 3.976528 | 192.168.1.49 | 192.163.96.1 | TCP    | 60  | 49809 → 13 [PSH] Seq=0 Win=4048 Len=0 MSS=1460 WS=0 SACK_PERM  |
| 40 | 4.139237 | 132.163.96.1 | 192.168.1.49 | TCP    | 66  | 13 → 49809 [SYN, ACK] Seq=0 Ack=1 Win=5535 Len=0 MSS=1460 SACK_PERM                                  |
| 41 | 4.139383 | 192.168.1.49 | 132.163.96.1 | TCP    | 54  | 49809 → 13 [ACK] Seq=1 Ack=1 Win=131328 Len=0  |
| 42 | 4.200753 | 132.163.96.1 | 192.168.1.49 | DHTIME | 166 | DHTIME Response  |
| 43 | 4.200753 | 132.163.96.1 | 192.168.1.49 | TCP    | 60  | 13 → 49809 [FIN, ACK] Seq=52 Ack=1 Win=6816 Len=0  |
| 44 | 4.299808 | 192.168.1.49 | 132.163.96.1 | TCP    | 54  | 49809 → 13 [ACK] Seq=1 Ack=53 Win=131328 Len=0   |
| 45 | 4.380883 | 192.168.1.49 | 132.163.96.1 | TCP    | 54  | 49809 → 13 [PSH, ACK] Seq=1 Ack=53 Win=131328 Len=0  |
| 46 | 4.465541 | 132.163.96.1 | 192.168.1.49 | TCP    | 60  | 13 → 49809 [ACK] Seq=53 Ack=2 Win=6752 Len=0   |
| 48 | 5.013487 | 192.168.1.49 | 192.168.1.39 | TCP    | 164 | 49775 → 8089 [PSH, ACK] Seq=111 Ack=111 Win=507 Len=110 [TCP segment of a reassembled PDU]           |
| 49 | 5.014220 | 192.168.1.39 | 192.168.1.49 | TCP    | 164 | 8089 → 49775 [PSH, ACK] Seq=111 Ack=221 Win=279 Len=110 [TCP segment of a reassembled PDU]           |
| 51 | 5.059924 | 192.168.1.49 | 192.168.1.39 | TCP    | 54  | 49775 → 8089 [ACK] Seq=221 Ack=221 Win=513 Len=0   |

Flags a tenir en compte:

**ACK:** Confirmar la recepció del missatge

**FIN:** Identificar a un paquet com l'últim de la connexió

**SYN:** Sincronitzar els nombres de seqüència inicials

El diagrama de seqüència de l'intercanvi de control TCP és el següent:

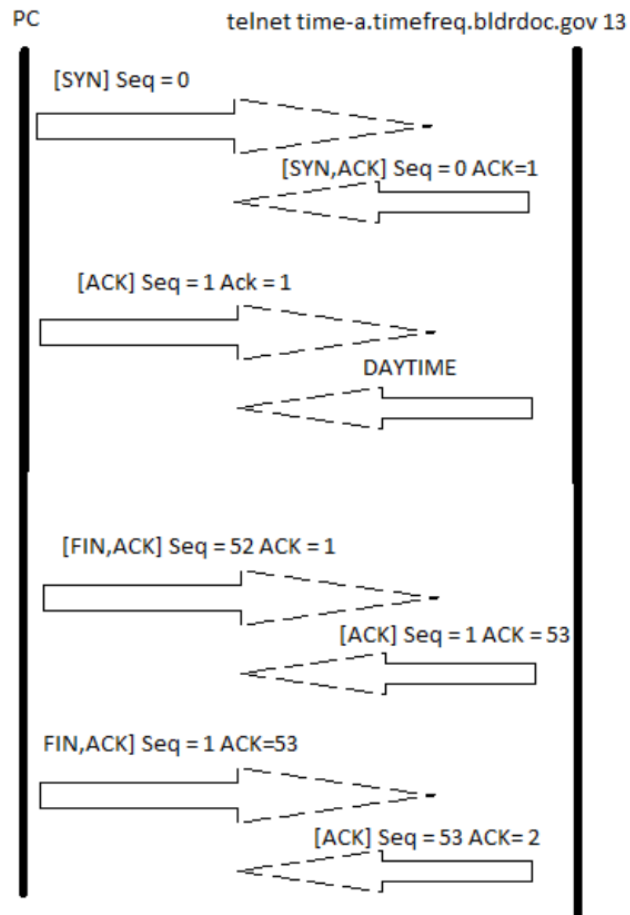


Figura 3.3: Diagrama seqüència intercanvi de control TCP

#### 4. EXERCICI 3

Ara moniteritzarem el trànsit de la nostra xarxa local quan feim ping a una adreça pública:

```

C:\Users\usuario>ping www.google.com

Haciendo ping a www.google.com [142.250.200.68] con 32 bytes de datos:
Respuesta desde 142.250.200.68: bytes=32 tiempo=15ms TTL=116
Respuesta desde 142.250.200.68: bytes=32 tiempo=14ms TTL=116
Respuesta desde 142.250.200.68: bytes=32 tiempo=15ms TTL=116
Respuesta desde 142.250.200.68: bytes=32 tiempo=14ms TTL=116

Estadísticas de ping para 142.250.200.68:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 14ms, Máximo = 15ms, Media = 14ms
  
```

Per buscar dins del programa podrem filtrar amb l'adreça que hem fet ping per poder obtenir

nomes els resultats desitjats. Estaran utilitzant el protocol ICMP, que serveix per a l'enviament de missatges d'error o èxit en connectar-se amb una altra adreça IP.

| No. | Time      | Source         | Destination    | Protocol | Length | Info   |
|-----|-----------|----------------|----------------|----------|--------|--|
| 149 | 24.937584 | 192.168.1.49   | 142.250.200.68 | ICMP     | 74     | Echo (ping) request id=0x0001, seq=13/3328, ttl=128 (reply in 150) |
| 150 | 24.952603 | 142.250.200.68 | 192.168.1.49   | ICMP     | 74     | Echo (ping) reply id=0x0001, seq=13/3328, ttl=116 (request in 149) |
| 156 | 25.954930 | 192.168.1.49   | 142.250.200.68 | ICMP     | 74     | Echo (ping) request id=0x0001, seq=14/3584, ttl=128 (reply in 163) |
| 163 | 25.969659 | 142.250.200.68 | 192.168.1.49   | ICMP     | 74     | Echo (ping) reply id=0x0001, seq=14/3584, ttl=116 (request in 156) |
| 170 | 26.959387 | 192.168.1.49   | 142.250.200.68 | ICMP     | 74     | Echo (ping) request id=0x0001, seq=15/3840, ttl=128 (reply in 171) |
| 171 | 26.974296 | 142.250.200.68 | 192.168.1.49   | ICMP     | 74     | Echo (ping) reply id=0x0001, seq=15/3840, ttl=116 (request in 170) |
| 177 | 27.974744 | 192.168.1.49   | 142.250.200.68 | ICMP     | 74     | Echo (ping) request id=0x0001, seq=16/4096, ttl=128 (reply in 178) |
| 178 | 27.989564 | 142.250.200.68 | 192.168.1.49   | ICMP     | 74     | Echo (ping) reply id=0x0001, seq=16/4096, ttl=116 (request in 177) |

Ara, podem desglossar totes les dades que conté:

```
Internet Protocol Version 4, Src: 192.168.1.49, Dst: 142.250.200.68
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 60
    Identification: 0x54df (21727)
  > 000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 128
    Protocol: ICMP (1)
    Header Checksum: 0x0000 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 192.168.1.49
    Destination Address: 142.250.200.68
```

Figura 4.1: Dades protocol ICMP

**Type:** 0 si és una resposta o 8 si és una petició.

**Code:** Byte que sempre serà 0.

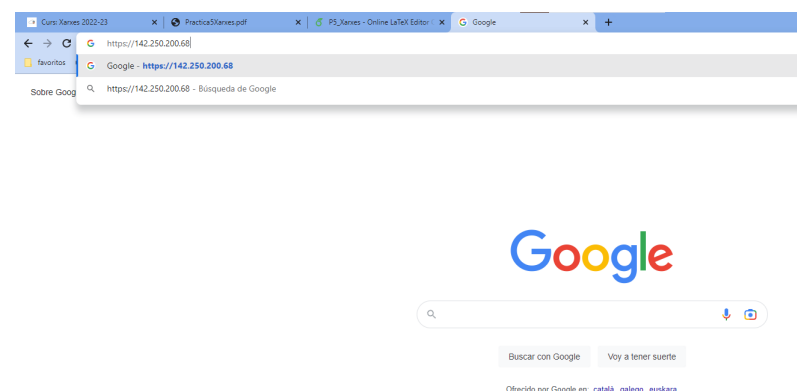
**Checksum:** Detecció d'errors.

**Identifier (BE) i Sequence Number (BE):** Identificador (escrit en format Big Endian) per comprovar si la dada és la que estem esperant.

**Identifier (LE) i Sequence Number (LE):** Identificador (escrit en format Little Endian) per comprovar si la dada és la que estem esperant.

**Data:** Payload de la trama.

I si busquem la IP destí al nostre cercador ens envia a [www.google.com](http://www.google.com).



El sniffer captura tot l'intercanvi d'informació entre el nostre PC i el servidor remot, que es realitza mitjançant els protocols UDP i TCP, que s'usaran per a l'enviament de paquets i l'establiment



de la connexió, respectivament.

El protocol TCP usa per defecte el port 443 per a realitzar transferències d'hipertext:

```

Transmission Control Protocol, Src Port: 49896, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
  Source Port: 49896
  Destination Port: 443
  [Stream index: 8]
  [Conversation completeness: Incomplete (12)]
  [TCP Segment Len: 1]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 953609980
  [Next Sequence Number: 2 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 839893948
  0101 .... = Header Length: 20 bytes (5)

```

Figura 4.2: Dades protocol TCP

Aquest protocol s'usa per la comunicació i utilitza els següents paquets on les seves funcions són:

**SYN:** Sincronitzar els números de seqüència inicials.

**SYN/ACK:** Validar la sincronització realitzada anteriorment.

**ACK:** Confirmar l'enviament de tots els anteriors missatges.

**FIN,ACK:** Per finalitzar l'enviament.

## 5. EXERCICI 4

Al quart exercici, utilitzarem **Packet Tracer** en ,és profunditat, en aquest exercici crearem dues xarxes, una xarxa estarà formada per IPs estàtiques configurades manualment i l'altra per DHCP i, a continuació, una de les xarxes estarà connectada a un servidor. En el nostre cas hem connectat el servidor a la xarxa estàtica i les dues últimes xarxes estaran connectades entre si mitjançant el núvol.

El diagrama de la topologia de la xarxa és el que es mostra a la següent figura, indicant l'adreça IP de cada ordinador i del servidor.

Per garantir la correcta connectivitat entre les dues xarxes a través del núvol, fem servir el PCO de la xarxa **DHCP** per enviar un paquet a l'ordinador PC4 de la xarxa estàtica. A la figura podem observar que les dues xarxes es poden connectar.

El mateix mètode s'utilitza per provar la comunicació entre PC0 i el servidor i permetre que el servidor envii una sol·licitud de resposta. A partir dels resultats de les següents figures, podem veure que l'ordinador i el servidor es poden comunicar entre ells sota diferents xarxes.



| Fire  | Last Status | Source | Destination | Type | Color   | Time(sec) | Periodic | Num | Edit   | Delete   |
|---|-------------|--------|-------------|------|---|-----------|----------|-----|--------|----------|
|  | Successful  | PC0    | Server0     | ICMP |  | 0.000     | N        | 0   | (edit) | (delete) |

Figura 5.1: PC a server

Mitjançant les proves anteriors, hem garantit les funcions de comunicació entre ordinadors i servidors sota diferents xarxes, a continuació, provarem les funcions de les pàgines web del

| Fire | Last Status | Source  | Destination | Type | Color | Time(sec) | Periodic | Num | Edit   | Delete   |
|------|-------------|---------|-------------|------|-------|-----------|----------|-----|--------|----------|
|      | Successful  | Server0 | PC0         | ICMP |       | 0.000     | N        | 0   | (edit) | (delete) |

Figura 5.2: Server a PC

servidor a les quals accedeixen els ordinadors.

Com es pot veure en les imatges anteriors, podem utilitzar perfectament el lloc web del servidor al qual accedeix l'ordinador de PC0. A continuació introduïrem alguns frames i protocols entre la comunicació PC-servidor. En primer lloc, podem trobar que la comunicació entre l'ordinador i el servidor utilitza el protocol **TCP**.

Segons la nostra anàlisi de **Flag TCP** al segon exercici, podem trobar que la primera sol·licitud iniciada pel PC al servidor és una sol·licitud **SYN**.

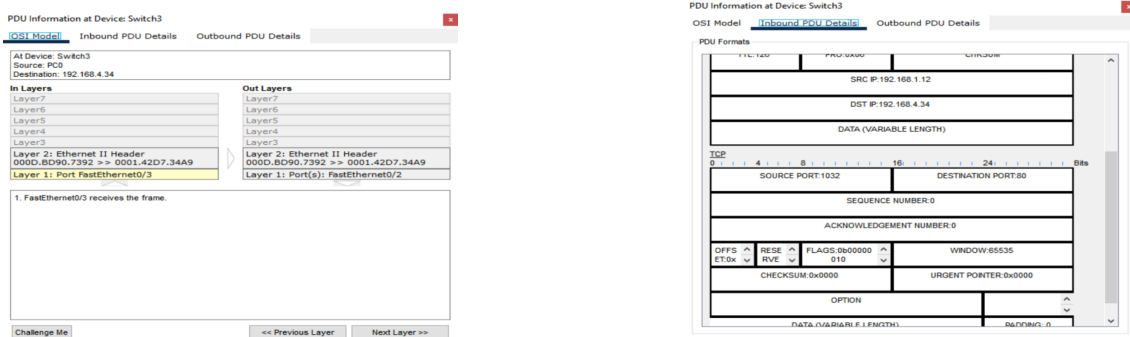


Figura 5.3: PDU PC request

De la mateixa manera, segons la nostra anàlisi del **Flag TCP** al segon exercici, podem trobar que una resposta del servidor al PC és una resposta **ACK**.

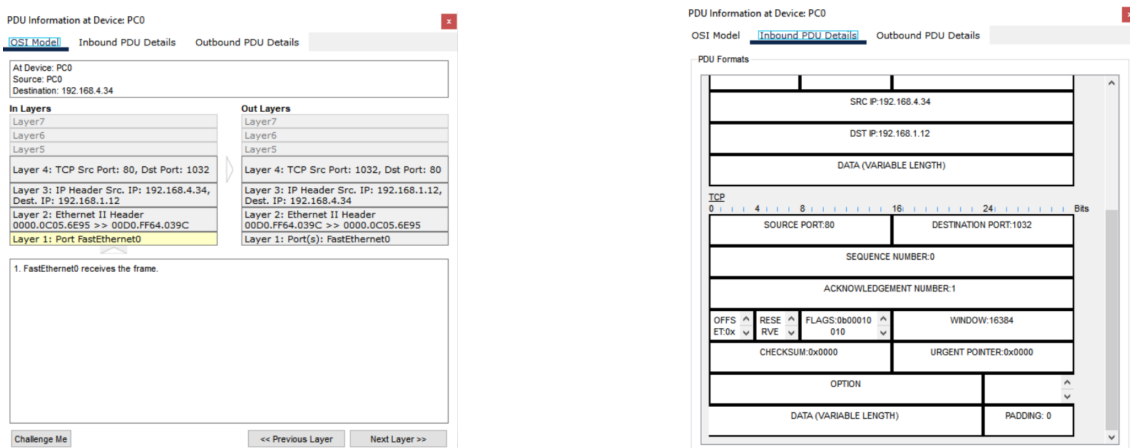


Figura 5.4: PDU server response

També podem analitzar altres paquets de petició de la mateixa manera, però aquí analitzem un protocol que no s'ha vist al segon exercici.

Aquí podem veure el protocol **HTTP**, el protocol de transferència d'hipertext estableix el protocol per a l'intercanvi de documents d'hipertext i multimèdia al web. **HTTP** disposa d'una variant xifrada mitjançant **SSL** anomenada **HTTPS**.

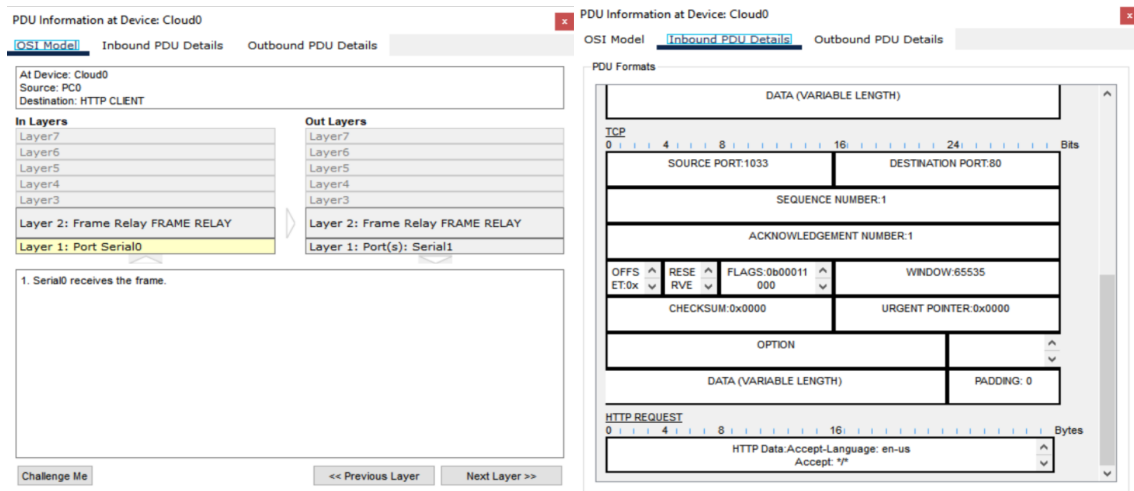


Figura 5.5: PDU HTTP request

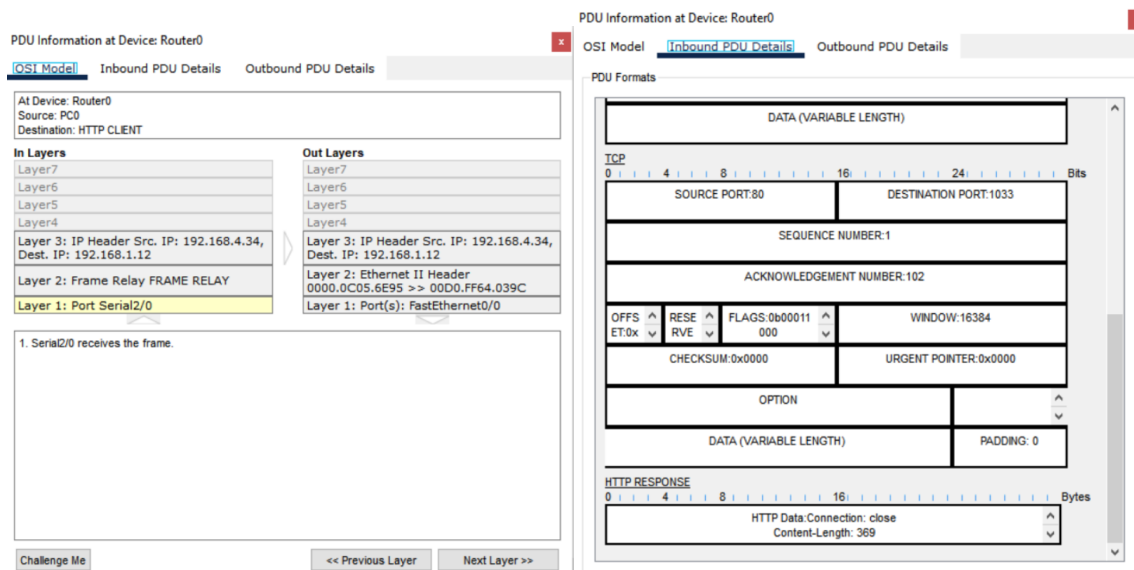


Figura 5.6: PDU HTTP response

Aquí podem veure que el protocol **TCP/IP + HTTP** s'utilitza quan l'ordinador i el servidor estableixen una sol·licitud web de la connexió. El que hem de prestar atenció aquí és que quan l'ordinador envia una sol·licitud web al servidor, fa servir **HTTP request**, i el servidor utilitza **HTTP response** quan vol que l'ordinador envii una resposta.

Per tal d'introduir d'una forma més clara la comunicació PC-servidor, realitzem un diagrama per presentar el procés de comunicació esmentat. Per tant observem a la següent figura que, quan l'ordinador vulgui sol·licitar la web des del servidor, primer enviarà una sol·licitud de **SYN** al servidor, després el servidor respondrà a la sol·licitud (**SYN ACK**) i l'ordinador també respondrà **ACK**, assegurant així l'establiment de la comunicació entre l'ordinador i el servidor. Quan

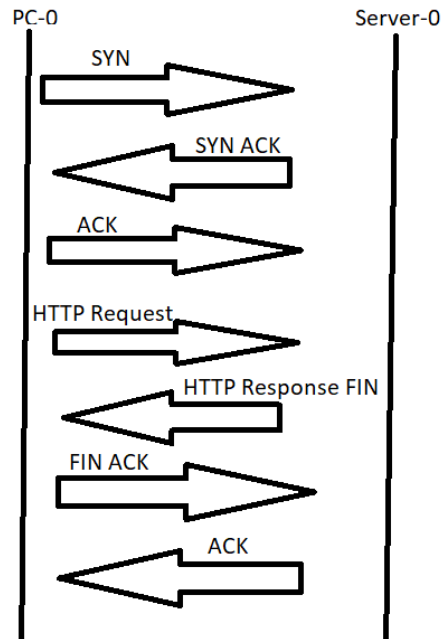


Figura 5.7: Diagrama de petició

l'ordinador confirmi establir una sol·licitud amb el servidor, enviarà una sol·licitud **HTTP request**, després quan el servidor rebí la sol·licitud, retornarà una altra de tipus **HTTP response**, un component principal del web. I amb una sol·licitud d'alerta (FIN), vol dir que la sol·licitud **HTTP** s'ha completat. Quan l'ordinador rebí les peticions **HTTP response** i FIN enviades pel servidor, finalment demanarà al servidor que tanqui la comunicació, quan el servidor rebí la sol·licitud de tancament de la comunicació, tancaran oficialment la sessió.

## 6. CONCLUSIONS

Aquesta pràctica ens ha servit molt per a acabar d'aprofundir en els conceptes estudiats i entendre en profunditat els diferents protocols esmentats a la pràctica. Hem après a utilitzar l'eina d'anàlisi de xarxa Wireshark i hem analitzat la xarxa tenint una comprensió més profunda amb el model OSI. A més, també hem dissenyat i interconnectat diverses xarxes a través del núvol en Packet Tracer.