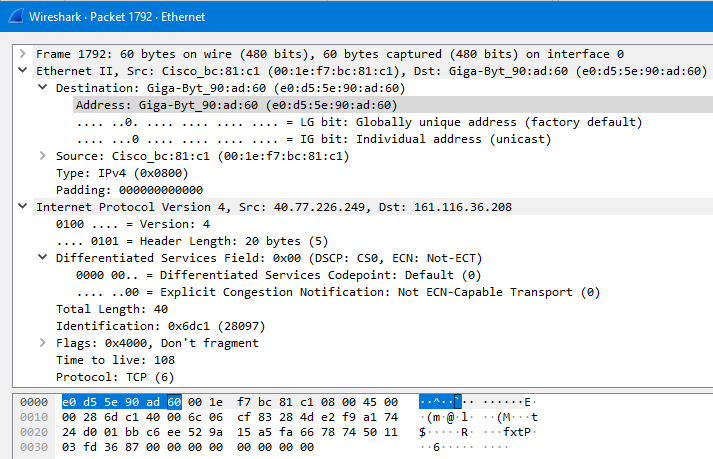
|  |
| --- |
| Francisco Díaz Ruiz NIUB 16828405  David Martín Vilar NIUB 16843606 |

|  |
| --- |
| Universitat de Barcelona |
| Pràctica 4: El model OSI |
| Xarxes |

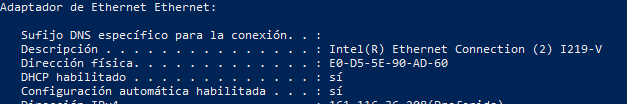
* **Objectius**

Els objectius de la pràctica són aprendre com funciona l’encapsulament de diferents Unitats de Protocol d’Usuari (DPU) així com a veure com funciona Wireshark.

* **Exercici 1:**



Així apareix la nostre adreça MAC al Wireshark. Mentre que amb la comanda “ipconfig /all” apareix de la següent manera:



Com es pot observar, la diferencia entre la MAC donada per el Wireshark i la que es proporcionada amb la comanda de terminal són quasi iguals, la única diferencia és que el Wireshark també proporciona la MAC del equip receptor.

* **CAMPS DE LA IPv4:**
  + **Versió (4 bits):** Compresa entre el bit 0 i el bit 3, descriu el format de la capçalera utilitzada.
  + **Mida Capçalera (4 bits):** Compresa entre els bits 4 i 7, aquesta representa la longitud de la capçalera en paraules de 32 bits, el valor mínim és de 5 paraules, és a dir, de 20 Bytes i màxim 15 paraules (60 Bytes).
  + **Tipus de Servei (8 bits):** Comprés entre els bits 8 i 15, aquest camp indica una sèrie de paràmetres sobre la qualitat del servei desitjada durant la transmissió per una xarxa. Depenent de la xarxa ofereix diferents nivells de prioritats als serveis i, per tant, podem dividir els bits, aquesta divisió és la següent:
    - Els 3 primers bits estan relacionats amb la procedència dels missatges, juntament amb un indicador que indica el nivell d’urgència següents:

000: De rutina.

001: Prioritari.

010: Immediat.

011: Llamp.

100: Invalidació llamp.

101: Processant trucada crítica i d’emergència.

110: Control de treball d’Internet.

111: Control de xarxa.

* Els darrers 5 bits de menor pes són totalment independents i indiquen les característiques del servei.

**Aquests 3 darreres divisions de la IP formen altres camps, aquests són:**

* + - La versió, juntament amb la mida de la capçalera i el tipus de servei formen l’identificador de la direcció IP. Aquest identificador s’utilitzarà en cas de que el datagrama que identifica hagi de ser fragmentat per així distingir els fragments entre altres fragments d’altres datagrames.
    - Tanmateix, la versió i la mida formen el temps de vida mentre que el tipus de servei representa el protocol.
  + **Longitud Total (16 bits):** Aquest cap està comprès entre els bits 16 i 31 de la IP. Indica la mida total del datagrama, incloses les mides de la capçalera i la de les dades. En cas de que el datagrama s’hagués fragmentat, aquest cap contindrà la mida del fragment i no del datagrama original.

**Tanmateix aquest últim cap es pot dividir en dos camps diferents:**

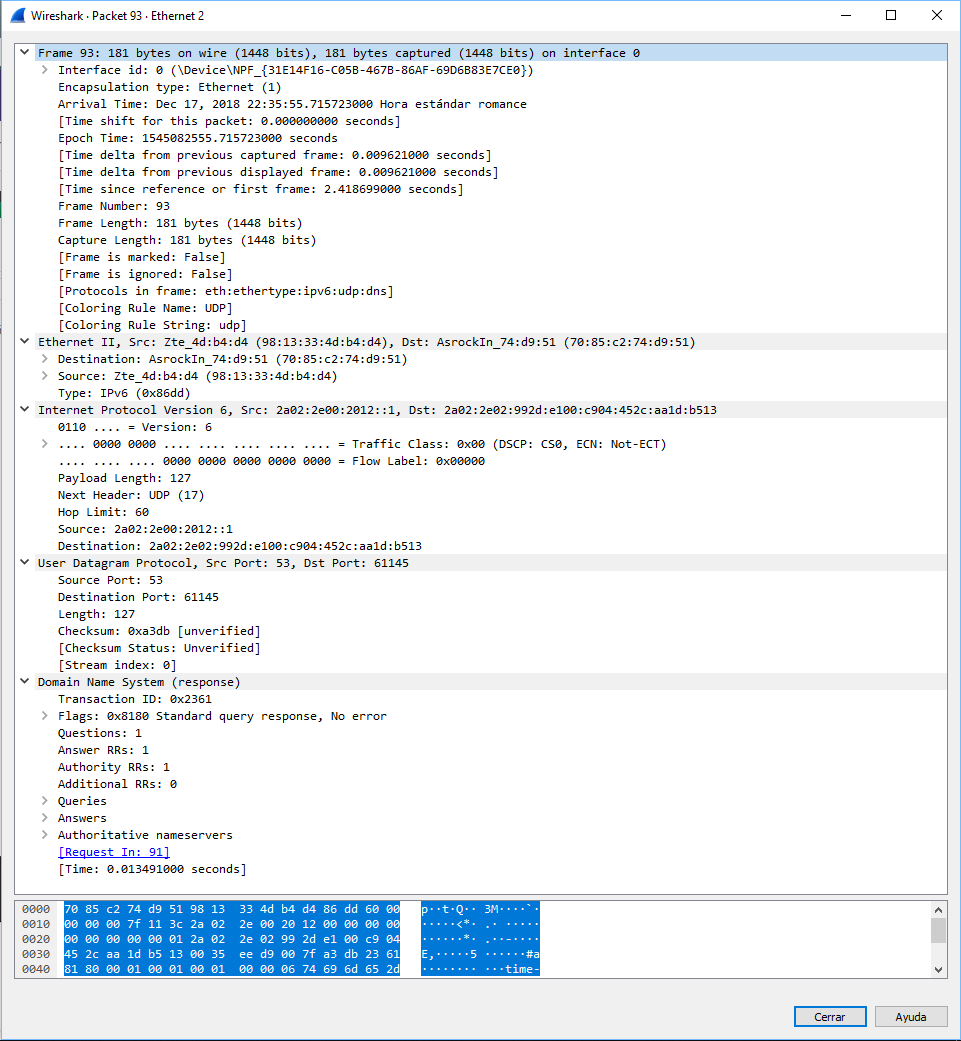
* + - **Flags (3 bits):** Comprès entre els bits 16 i 18, s’utilitza només per especificar els valors relatius de la fragmentació dels paquets, els 3 bits són:
      * bit 0: Reservat; ha de ser 0
      * bit 1: 0 = Divisible, 1 = No Divisible (DF)
      * bit 2: 0 = Últim Fragment, 1 = Fragment Entremig (el segueixen més fragments) (MF)
    - **Posició del Fragment (13 bits):** Aquests indiquen en paquets fragmentats la posició que ocupa el paquet actual dintre del datagrama original, i sent el primer paquet d’una sèrie de fragments amb valor 0.
* **Exercici 2:**

Després d’observar l’execució de la comanda ‘telnet time-A.timefreq.bldrdoc.gov 13’ es pot veure que intenta connectar-se a un port, però com que no hem introduït cap IP dins la comanda, perd la connexió amb el host i es desconnecta. Mentrestant, en el Wireshark es pot observar com apareixen la nostra IP i l’altra, són IPv6.



La seva IP és: 2a02:2e00:992D:e100:c904:452c:aa1d:b513

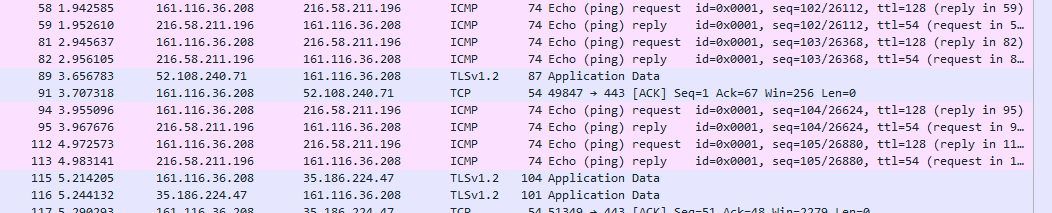
Entrem al paquet:



El port que ens apareix dintre del Wireshark al que ens volem connectar es el 61145

Podem observar que el nostre equip té en aquest instant el port 53 com a source i es vol connectar al port 61145 que hem mencionat abans. També es pot veure que estem utilitzant el protocol UDP per realitzar aquesta connexió.

* **Exercici 3:**

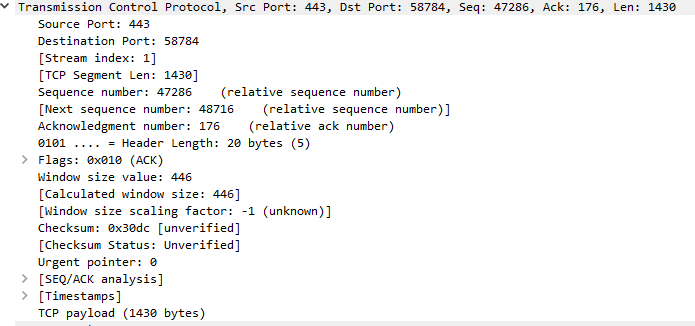


Com es pot observar en la captura, l’ordinador ha executat el *ping* que hem demanat, en aquest cas a la pàgina web ‘www.google.com’ i ha retornat aquests valors. S’ha connectat al port 216.58.111.196 utilitzant el protocol ICMP. Tant el *Echo request* com el *Echo reply* utilitzen el protocol mencionat per tal de realitzar la comunicació entre els dos ports.

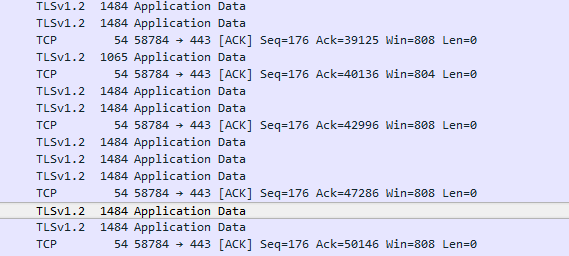
El protocol ICMP (Internet Control Message Protocol) s’utilitza directament dins de l’arquitectura TCP/IP i la seva funció és observar e informar de l’estat i situacions d’error en el funcionament de la capa de xarxa de l’arquitectura, centrat en aspectes com l’encaminament, fragmentació, congestió de dades, etc.

Els camps que conté aquest protocol són: Tipus, codi, Checksum i informació variable.

Una vegada ens connectem a la IP obtinguda amb el ping a la pàgina web, podem observar que el sniffer ha capturat totes les transmissions de paquets entre les dues IPs, on aquestes són la de la pàgina web i la del nostre ordinador. Com a descripció del que ha capturat l’sniffer, ens mostra que es application data i que utilitza el protocol TCP per connectar-se amb el servidor de la pàgina web on hem fet prèviament el ping. Respecte a si s’obre la pàgina, una vegada hem introduït la direcció “http://ip\_obtinguda” ens entra dintre de la web sense cap problema i la podem utilitzar com quan s’utilitza normalment.



En la captura de l’apartat del TCP del resultat obtingut mitjançant el sniffer es pot observar les diferents dades que s’obtenen o s’utilitzen durant la connexió. Respecte al port que estem utilitzant per connectar-nos a la pàgina web aquest seria el port 58784 (en la captura surt com a Destination port) mentre que el port on ens volem connectar es el port 443 (en la captura apareix com a Source port). Aquesta captura mostra la recepció de dades per part del nostre ordinador.

En la connexió entre les dues IPs es pot veure que s’utilitzen protocols de control de flux, en aquest cas es pot veure que mentre un dels dos envia dades l’altre està a l’espera mentre arriben les dades, indicant que s’utilitza un protocol de control de flux Full dúplex típic del protocol TCP.

En aquesta captura podem observar el que ens apareix al sniffer mentre s’està efectuant la connexió a la pàgina, en aquest cas la pàgina a la que ens hem connectat es la pàgina principal de Google ([www.google.com](http://www.google.com)). Respecte a les dades que es poden observar en la captura, veiem que el port de la web envia Application Data al nostre equip i una vegada aquest les rep envia un ACK indicant que han sigut rebudes adequadament.

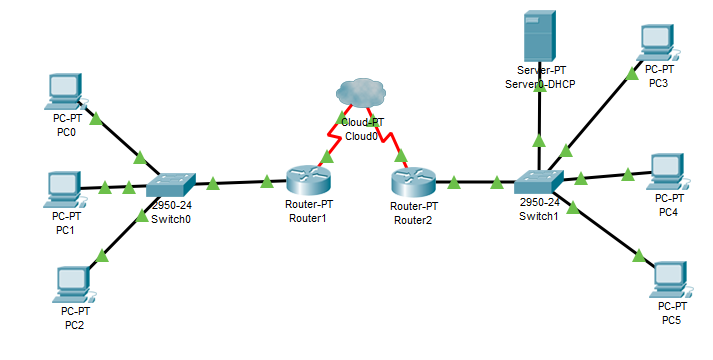
* **Exercici 4**

Les dues xarxes funcionen per separat però no hem pogut fer la connexió via cloud entre ambdues xarxes.

La xarxa del router 1 utilitza subnetting variable amb IPs estàtiques, la xarxa del router 2 té un servidor DHCP per assignar les IP.

La primera xarxa es compon del router, un switch y 3 ordinadors, el funcionament es com el de la xarxa de la pràctica anterior.

La segona xarxa està composada del router, un switch, un servidor y 3 ordinadors. Està configurat perquè el servidor, que es DHCP faci l'assignació de les adreces.



No hem pogut fer funcionar la connexió entre ambdues xarxes per tant no podem fer les comunicacions entre elles.

* **Conclusions i observacions:**

En conclusió, hem observat com s’envien les dades entre diferents xarxes i, tanmateix, com funciona els protocols TCP, DPU, DNS i ICMP. Cal dir que hem tingut problemes a l’hora de realitzar uns quants d’aquests exercicis. Per exemple, a l’hora de realitzar l’exercici 2, depenent com no sortia al sniffer el protocol DNS que ens tenia que indicar a quina IP ens havíem de connectar posteriorment a l’execució de la comanda de Telnet.