Xarxes
Pràctica 4. El model OSI

Informe realitzat per Joaquím Yuste i Marcos Plaza

Universitat de Barcelona
Facultat de Matemàtiques i Informàtica

Índex

Objectius de la pràctica

L'objectiu primordial de la pràctica és conèixer més en profunditat l'encapsulació de les diferents Unitats de Protocol d'Usuari o DPU utilitzades en el model de referència OSI i en el protocol TCP/IP. Veurem quines són les peces primordials durant una connexió mitjançant el programa sniffer gratuït Wireshark. A més farem servir el software de simulació d'una xarxa vist amb anterioritat; Cisco Packet Tracer. Com sempre, analitzarem quins són els subobjectius que ens calen assolir per a dur a terme aquesta pràctica mitjançant els exercicis proporcionats a l'enunciat.

0) Primer ens centrarem en el funcionament i la interfície gràfica del programa Wireshark seguint una petita introducció. Donarem detalls sobre el nou funcionament del programa i les diferents decisions que hem pres, ja que hem fet servir una versió més actualitzada en comparació a la versió que s'ha fet servir per l'enunciat.

1) A continuació farem servir aquests coneixements previs per a realitzar l'exercici 1. Veurem com realitzar filtratge per IP dins Wireshark i analitzarem el conjunt de dades dins les adreces MAC i IP (tant IPv4 com IPv6) del nostre ordinador.

2) En el següent exercici veurem com es realitzen diferents connexions fent servir servidors externs. Veurem com funcionen dos dels protocols més utilitzats i quina relació tenen amb TCP/IP i el model OSI.

3) En aquest exercici ens disposarem a continuar analitzant altres protocols com l'utilitzat pel programa ping; l'ICMP. Explorarem el seu funcionament així com la seva utilitat dins la pila TCP/IP.

4) En l'últim problema plantejat farem servir el Cisco Packet Tracer per a establir dues xarxes (configurades de manera diferent entre si) i connectar-les al núvol. Combinarem Wireshark per a veure el tràfic generat per la connexió d'aquestes dues xarxes.

Reforçarem els exercicis amb alguns dels arxius de les captures realitzades a Wireshark, adjunts a aquest document a la carpeta anomenada 'wireshark captures'. A més per l'últim exercici adjuntarem els arxius corresponents a les xarxes muntades amb el Packet Tracer a la carpeta 'cisco packet tracer files'.

Realització de la pràctica

Abans d'entrar a realitzar els diferents exercicis, volem mencionar que la pràctica ha estat realitzada amb els dos programaris anteriorment esmentats; Wireshark i Cisco Packet Tracer. Cal tenir en compte que aquests dos programes no conserven del tot les característiques de versions anteriors les quals són utilitzades en l'enunciat de la pràctica.

En entrar a Wireshark veurem la següent finestra.

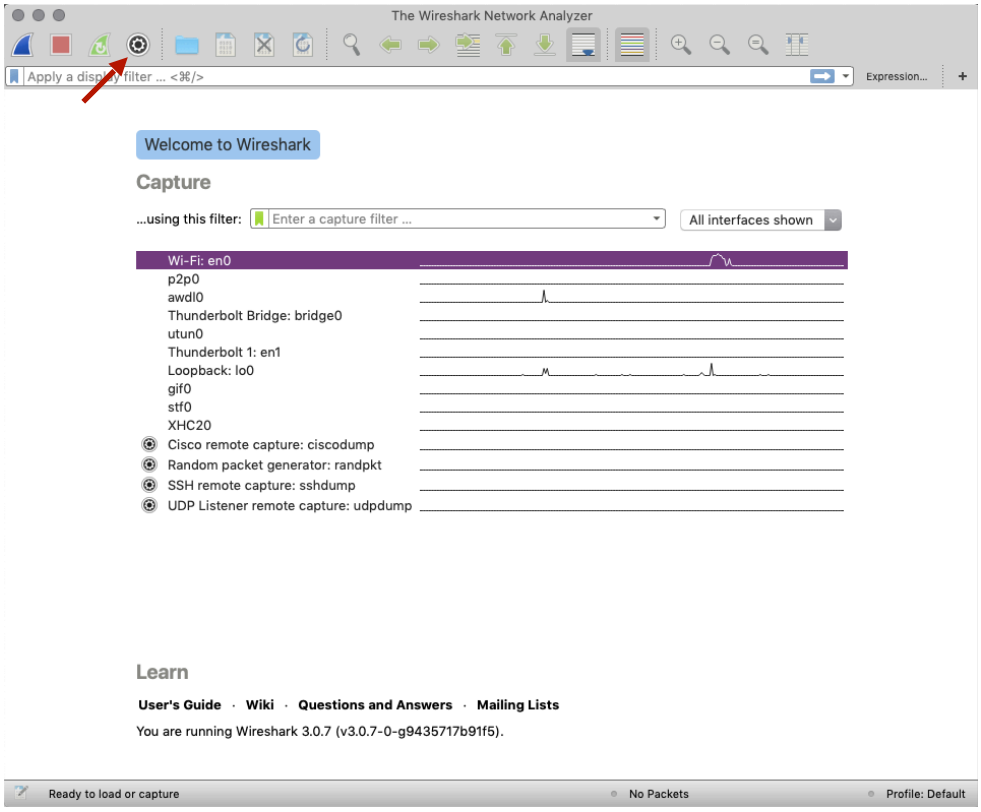


Figura 1. Finestra principal del programa Wireshark

Com podem observar apareix un llistat de totes les interfícies del sistema (tant les wireless, com les wired així com les 'externes'). En el nostre cas no apareix el botó 'details' que s'especifica a l'enunciat, ja que és una funció que apareix a versions més antigues del programa (tan sols en les versions de Windows). Per tant per a veure més detalls sobre les interfícies hem d'anar al botó apuntat per la fletxa vermella a la Figura 1 anterior ('Capture options'). Aquí podem veure la finestra següent.

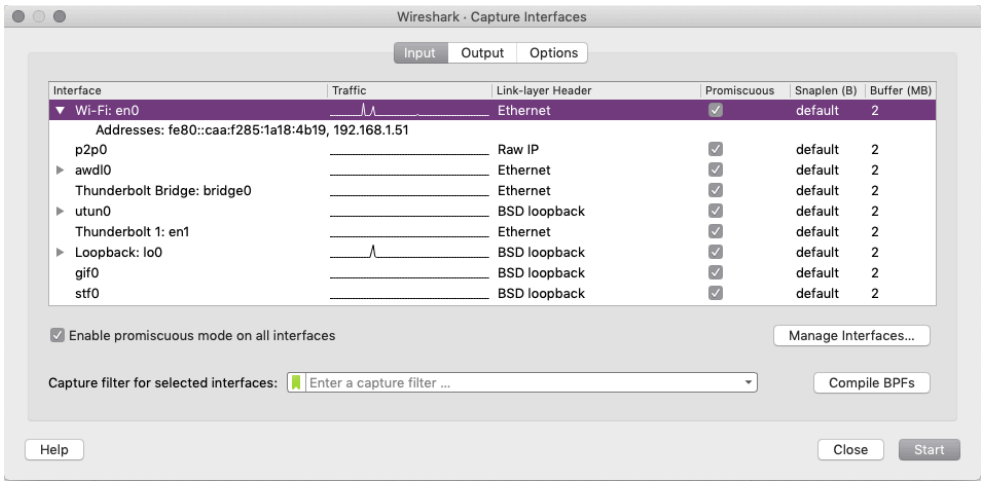


Figura 2. Finestra 'Capture options'

En seleccionar una de les interfícies del sistema, com per exemple la Wi-Fi, podem destacar diversos paràmetres. Observem que fa servir el port anomenat en0 (port Ethernet), a més també s'utilitza una gràfica senzilla per a descriure el tràfic de dades en aquesta interfície. Veiem també que aquest protocol que ens serveix per a connectar-nos a internet, fa servir dos tipus de adreces IP (IPv4 i IPv6). La IPv4 a l'equip on s'executa Wireshark és 192.168.1.51 mentre que la IPv6 és fe80::caa:f285:1a18:4b19. Per últim observem que Wi-Fi utilitza Ethernet en el paràmetre Link-layer Header.

Com en el nostre cas Wi-Fi utilitza Ethernet seleccionarem aquesta interfície. Si explorem una mica en la transmissió dels paquets de l'equip, apareix usualment les IP del nostre equip així com la del router al qual estem connectats. Si donem una ullada a un paquet on apareguin les dues respectives IP podem veure les adreçes MAC de tots dos dispositius.

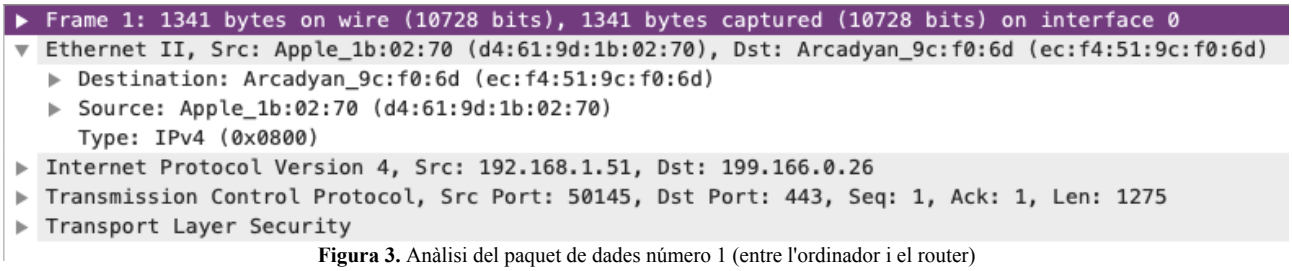


Figura 3. Anàlisi del paquet de dades número 1 (entre l'ordinador i el router)

Com podem veure en aquest cas estem enviant dades des de l'ordinador on realitzem la pràctica fins al router. Veiem també les dues MAC; la de l'ordinador és d4:61:9d:1b:02:70, mentre que la del router és ec:f4:51:9c:f0:6d. A continuació si fem un ifconfig veiem que es confirma la relació entre la IP origen o 'source' de la transmissió de l'anterior trama amb la de l'equip on s'executa Wireshark. També podem veure la MAC de l'equip en executar aquesta comanda.

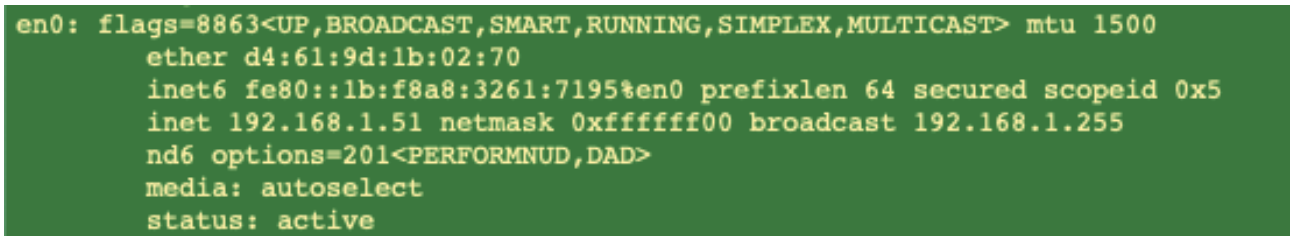


Figura 4. Execució de ifconfig. Ens fixem en el dispositiu 'en0' que s'encarrega de la connexió via Wi-Fi

Ara a la pàgina següent continuarem amb els exercicis proposats a la pràctica.

Exercici 1

Abans ja hem vist quina era la IPv4 i IPv6 del nostre equip (192.168.1.51). Aquest exercici diu que, el tràfic de dades que es dóna en un ordinador és d'un volum considerable. Així que ara es demana fer un filtratge d'aquells paquets que impliquin directament la nostra IP. Per a fer-ho farem servir la zona de filtratge de la finestra principal de Wireshark, tot introduint el següent; ip.addr == 192.168.1.51, en contraposició al ip.addr eq 192.168.1.51 que diu l'enunciat (recordem que a l'enunciat no es fa servir la versió més actualitzada de Wireshark).

A continuació seleccionarem un paquet on s'utilitzi el protocol TCP (és molt semblant al paquet analitzat a la Figura 3).

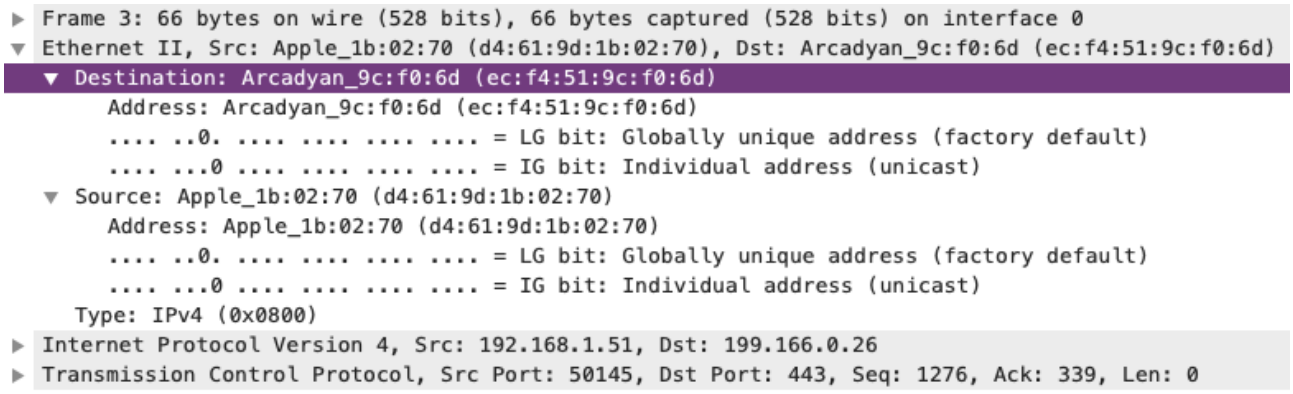


Figura 5. Anàlisi d'un paquet TCP qualsevol

Tal com ens ha passat abans observem tant la IP com la MAC del nostre equip així com la del destinatari. Com hem dit la nostra MAC és d4:61:9d:1b:02:70. Si ens fixem en la Figura 4 (on es fa un ifconfig) confirmem quina és l'adreça MAC del nostre equip.

Quan se'ns pregunta sobre les dues parts d'una adreça MAC, si fem una cerca per internet trobem que la primera meitat d'una adreça MAC indica el fabricant del dispositiu de maquinari. Els grans fabricants s'assignen més d'un codi. La segona meitat d'una adreça MAC indica el número de sèrie del dispositiu individual. Aquest és un número únic i també està assignat pel fabricant.
És responsabilitat del fabricant assegurar que tota l'adreça MAC és única per a cada producte individual.

D'altra banda, segons la Figura 5, podem veure que l'adreça MAC es desglossa en dues parts mitjançant l'LG bit i l'IG bit.
Tant el bit LG (de vegades també anomenat bit UL) com el bit IG es troben al byte més significatiu de cada adreça MAC, on el bit IG és el bit menys significatiu en aquest byte i el bit LG el segon menys significatiu.
El bit IG distingeix si l'adreça MAC és una adreça individual o de grup. En altres paraules, un bit IG de 0 indica que es tracta d'una adreça MAC unicast, un bit IG d'1 indica una adreça de transmissió multicast o de transmissió.
El bit LG o UL, d'altra banda, distingeix les adreces MAC assignades per proveïdors i assignades administrativament.

Ara donarem una ullada a la capçalera IP i analitzarem cadascun dels camps. A continuació apareix una captura del paquet TCP. A continuació seleccionarem un paquet on s'utilitzi el protocol TCP (és molt semblant al paquet analitzat a la Figura 3).

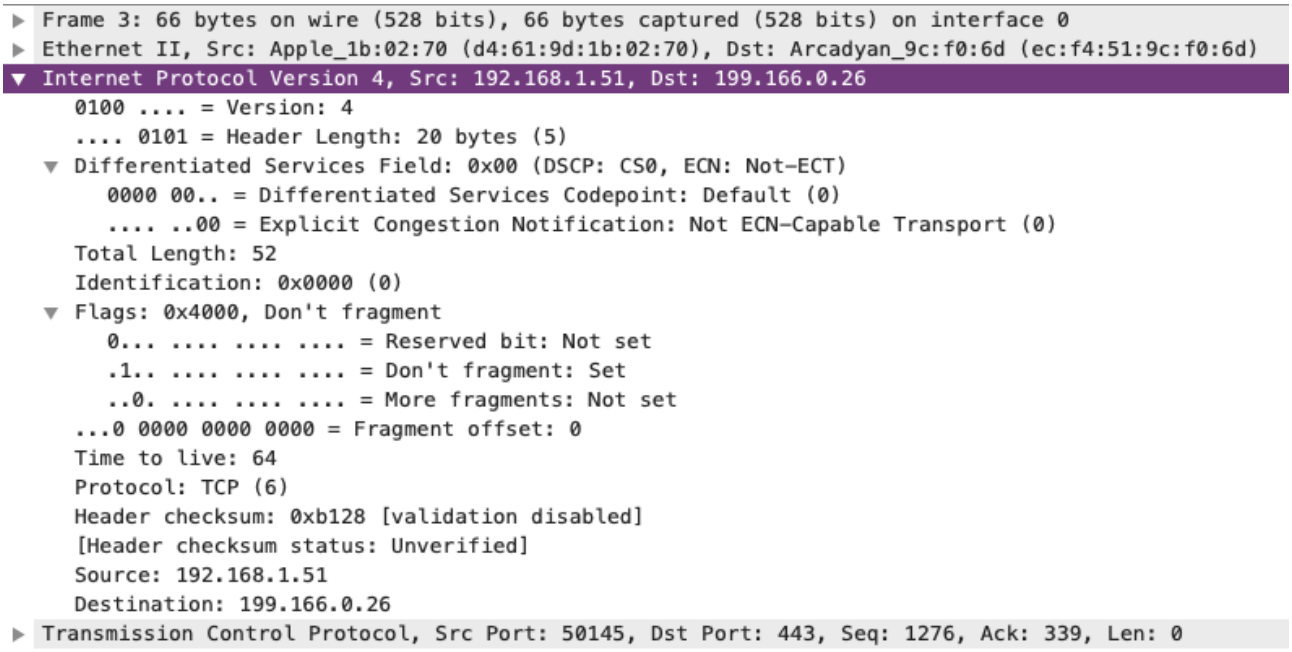


Figura 6. Paquet TCP anterior però amb la informació dels camps IP

Els camps de la capçalera IP que apareixen a Wireshark són els següents:

1) Versió de la IP. Correspon als primers 4 bits de la capçalera i ens indica si en la transmissió s'utilitza IPv4 o IPv6 (4 bits que corresponen al valor decimal 4 o 6 en binari, 0100 o 0110 respectivament).

2) Els següents 4 bits són la longitud de la capçalera, en paraules de 32 bits. El seu valor mínim és de 5 paraules (5x32 = 160 bits, 20 bytes) per a una capçalera correcta, i el màxim de 15 paraules (15x32 = 480 bits, 60 bytes).

3) Tipus de servei. Indica una sèrie de paràmetres sobre la qualitat de servei desitjada durant el trànsit per la xarxa. Algunes xarxes ofereixen prioritats de serveis, considerant determinat tipus de paquets "més importants" que d'altres. En total són 8 bits on els 3 primers indiquen la procedència del missatge i prenen valors del 0 al 7 en notació binaria. A mesura que augmentem de valor augmenta també el nivell de prioritat del paquet. Els altres 5 bits indiquen característiques independents entre si. Indiquen el grau de retard (normal o baix), el rendiment (normal o alt), la fiabilitat (normal o alta)... Els últims dos bits es deixen reservats.

4) És la mida total, en octets, del datagrama IP, incloent-hi la mida de la capçalera i el de les dades. En cas de fragmentació aquest camp contindrà la mida del fragment, no el del datagrama original.

5) Identificador únic del datagrama. S'utilitzarà, en cas que el datagrama hagi de ser fragmentat, per poder distingir els fragments d'un datagrama dels d'un altre.

6) Flags. Són 3 bits utilitzats per especificar valors relatius a la fragmentació de paquets. El bit 0 està reservat i sempre haurà de ser 0, d'altra banda el bit 1 ens indica si el paquet és divisible (0) o si el paquet no és divisible (1). Per últim el bit 2 indica si el fragment és l'últim en ser enviat (0).

7) Tot seguit als Flags tenim el 'Fragment offset' o posició de fragment. En paquets fragmentats indica la posició, en unitats de 64 bits, que ocupa el paquet actual dins el datagrama original. Per exemple, el primer paquet d'una sèrie de fragments contindrà en aquest camp el valor 0.

8) Els 8 següents són el temps de vida o TTL. Aquest paràmetre indica el màxim nombre d'encaminadors que un paquet pot travessar. Cada vegada que algun node processa aquest paquet disminueix el seu valor en, com a mínim, una unitat. Quan arribi a ser 0, el paquet serà descartat. Típicament pren el valor 64 o 128 en els datagrames.

9) En el byte següent trobem el protocol a utilitzar. Aquest camp indica el protocol de les capes superiors a què s'ha de lliurar el paquet. En el nostre és TCP, el qual està relacionat amb el número 6. Els altres protocols estan identificats amb un enter d'1 byte (de 0 a 255).

10) El camp següent és l'anomenat Checksum o suma de control. És un mètode de detecció d'errors en el datagrama i es recalcula cada vegada que algun node canvia algun dels seus camps (per exemple, el Temps de Vida).

11) Per últim trobem 64 bits que ens serveixen per referir-nos a la IP origen i destí, tal com podem veure a la Figura 6.

Amb això donem per finalitzat aquest primer exercici de la pràctica.

Exercici 2

En la realització d'aquest exercici es demana fer una captura de les comunicacions en executar la comanda telnet time-A.timefreq.bldrdoc.gov 13. Un cop executem a terminal, obtenim el següent.

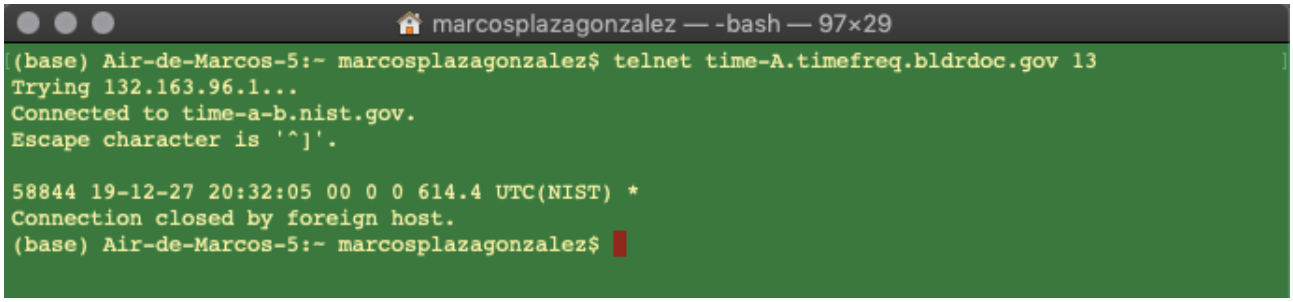


Figura 7. Output en executar la comanda telnet time-A.timefreq.bldrdoc.gov 13

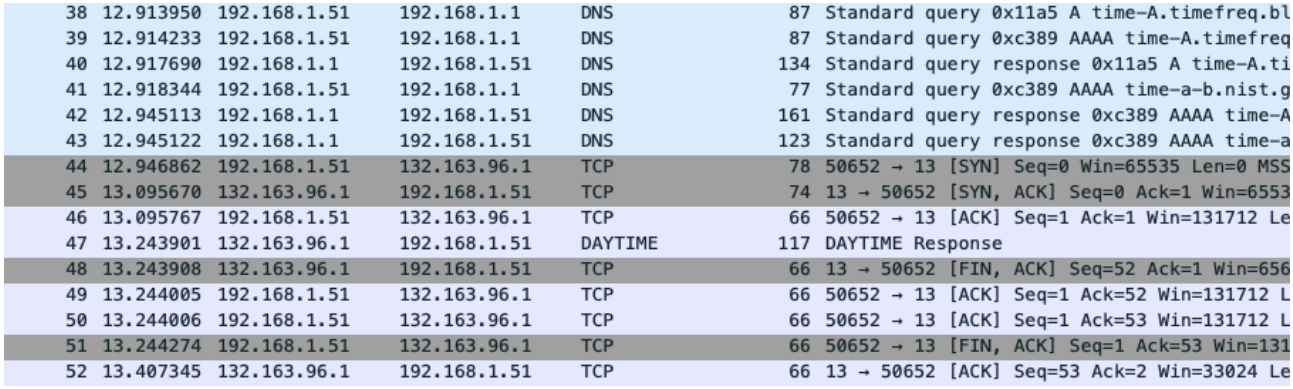


Figura 8. Tràfic de dades en executar la comanda telnet time-A.timefreq.bldrdoc.gov 13

Com podem apreciar a l'anterior comanda estem introduint el nom d'un domini. Per a poder-nos connectar al port i domini especificat haurem d'obtenir d'alguna manera la IP de la màquina a la qual ens volem comunicar. En aquesta tasca participa el protocol DNS, el qual s'encarrega de, a partir d'un nom de domini, establir la connexió amb el servidor DNS en qüestió, per a continuació obtenir (similar al funcionament d'una taula hash) la IP corresponent al nom de domini.

Si observem detingudament el port al qual ens estem connectant un cop coneguda la IP és el número 13 (ho indiquem en executar la comanda i posteriorment podem veure la resposta d'aquest a Wireshark). En aquesta ocasió estarem fent servir no únicament TCP (Transmission Control Protocol) com a protocol de transport, sinó també UDP (User Datagram Protocol).
Això és així, ja que a nivell de DNS estem fent servir aquest últim protocol de transport (UDP). El Domain Name System utilitza UDP, ja que l'objectiu és establir connexió amb el servidor DNS perquè aquest doni una resposta el més ràpid possible (query/response protocol), ja que a més ens interessa que aquest no estigui sobrecarregat de peticions. Per tant aquí la causa que DNS utilitzi UDP és essencialment l'estalvi de temps en proporcionar la IP relacionada al domini en particular (el protocol de transport TCP és més costós i pel que fa a velocitat és més lent).
Podem veure doncs, que la IP del servidor de DNS és la mateixa que la del nostre router, però aquí existeix la particularitat que UDP està fent servir el port de xarxa número 53 per a connectar-se al servidor. Per tant la IP del servidor de DNS al que ens estem connectant correspon a 192.168.1.1#53. En aquest cas la resposta s'especifica com una IP explicitament, la del domini al qual ens volem connectar 132.163.96.1.

A la següent pàgina podrem veure la Figura 9 corresponent al diagrama temporal d'intercanvi de paquets entre el nostre equip i el servidor DNS. Veurem que el nostre ordinador envia dos tipus de peticions. En un paquet enviem una petició del tipus A i a l'altre el tipus és AAAA.

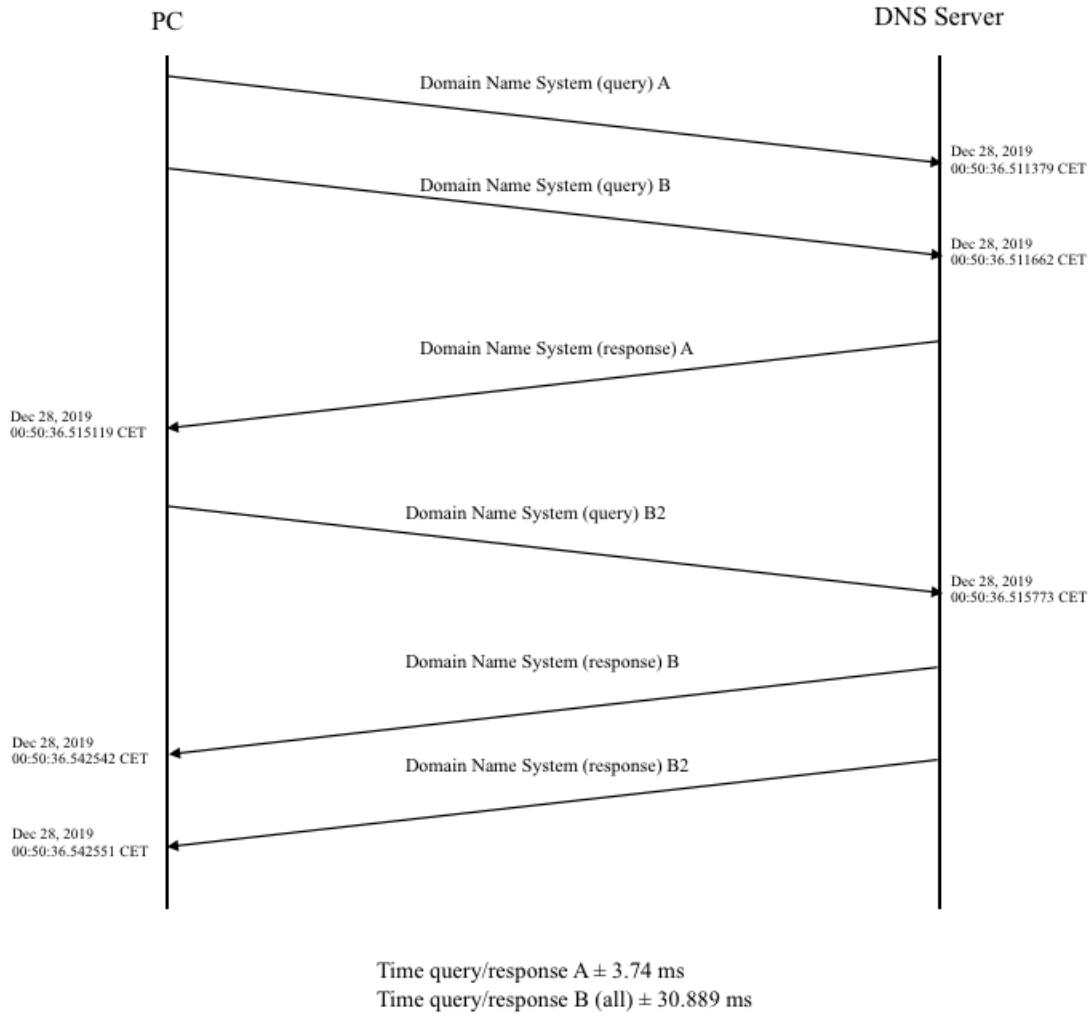


Figura 9. Diagrama temporal de la connexió amb el servidor DNS.

Cal mencionar que el query/response A correspon al tipus A, mentre que el B correspon al tipus AAAA anteriorment esmentat.
Dins del protocol DNS, existeixen diferents tipus de peticions o query. En aquest cas ens hem trobat amb les del tipus A i AAAA.
Les query de tipus A són peticions per obtenir adreces IP tipus IPv4, mentre que les del tipus AAAA ens serveixen per especificar que volem obtenir IPv6.
En aquest cas explicarem el tipus de petició tipus A. Inicialment volem connectar-nos al nom de domini especificat. El nostre sistema operatiu comprova la petició i veu que no té el nom de domini a memòria. Aleshores enviem un paquet tot especificant el domini del qual volem obtenir la resposta, en el nostre cas serà el domini A time-A.timefreq.bldrdoc.gov. Tot seguit el servidor DNS rebrà el paquet amb un temps de latència molt baix des de la petició. El servidor DNS (o un altre servidor encarregat de la [1]zona d'autoritat .gov) buscarà a la seva taula de dades de adreces IP, la IPv4 del nom de domini. Gairebé amb la mateixa velocitat el servidor de DNS contesta amb la IPv4 del domini anteriorment especificat. Si observem el diagrama temporal de pregunta i resposta des de el port del nostre equip fins al port del servidor DNS podem extreure que aquesta resposta és gairebé immediata, ja que triga un temps molt petit entre query i response (Per al query/respone tipus A és d'un temps de ± 3.74 ms). Això és degut en part a la utilització del protocol UDP dins la capa de transport del protocol DNS (basat en TCP/IP).

Com a últim punt d'aquest exercici, veurem l'intercanvi de control que es produeix a nivell de TCP per a la transmissió d'informació. Abans hem utilitzat un diagrama temporal per descriure la connexió entre el nostre ordinador i el servidor DNS. A la pàgina següent veurem un altre exemple de diagrama temporal, però aquesta vegada amb el servidor telnet (ja que ja sabem la seva IPv4 d'aquest).

[1] Servidors o grup de servidors que tenen assignats resoldre un conjunt de dominis (com el .es, .net, .cat o d'altres).

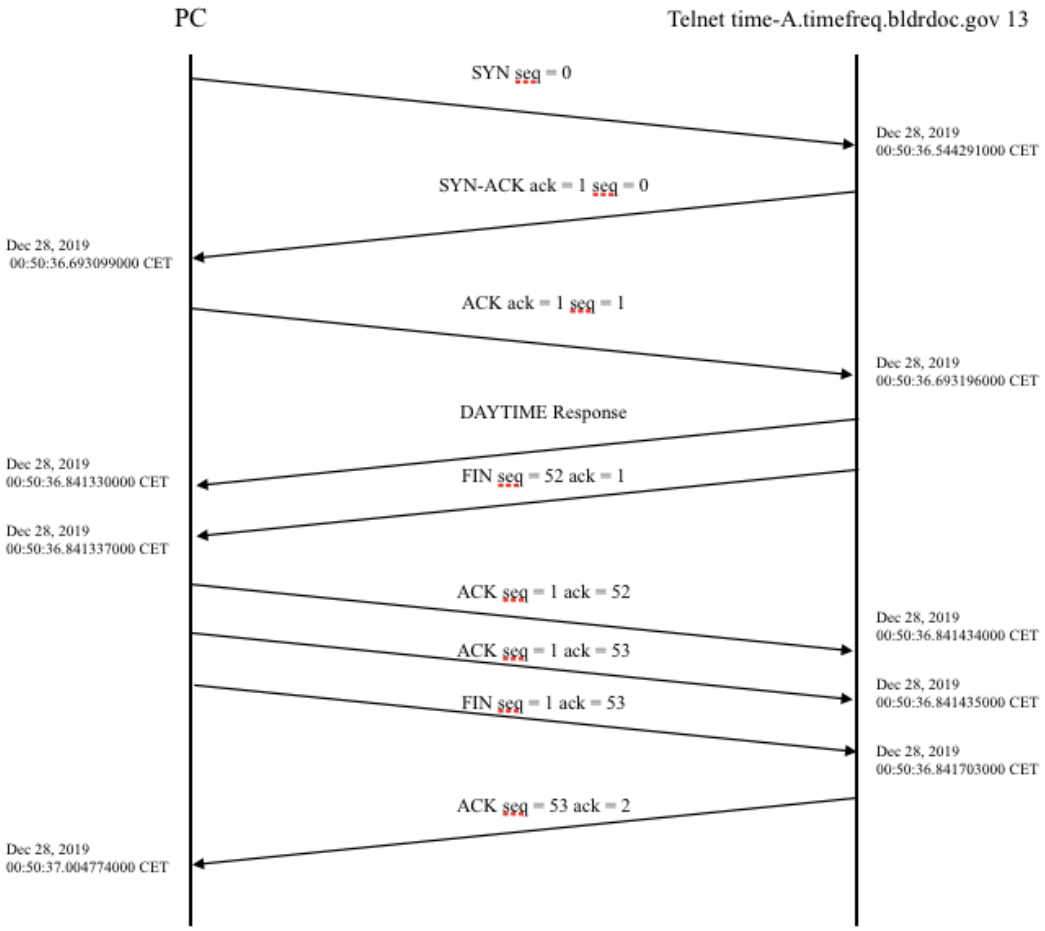


Figura 10. Diagrama temporal de la connexió amb el servidor Telnet.

En aquest diagrama ens centrarem en com establim la connexió amb el servidor Telnet. Dividirem l'explicació en tres passos molt senzills; establiment de la connexió, intercanvi de dades i finalització de la connexió.

Primerament, el que estem fent és establir la connexió amb el servidor Telnet mitjançant una metodologia anomenada negociació en tres passos. El nostre ordinador està realitzant una apertura activa d'un port TCP (en el nostre cas el 50652) i està enviant un paquet SYN per començar la negociació en tres passos. A continuació el servidor comprova que el port número 13 està obert perquè es pugui procedir a l'intercanvi d'informació. El servidor comunica al client que tot és correcte i respon a la petició SYN amb un paquet SYN/ACK. Finalment el client (nosaltres) estem contestant amb un ACK al servidor tot acabant la connexió en tres passos dient que estem llestos per a la comunicació.
Ara el servidor Telnet procedirà a utilitzar el DAYTIME Protocol el qual fa servir el port TCP/UDP 13 i envia la data i hora actuals com una cadena de caràcters.
Un cop hem rebut les dades que volíem, ara toca finalitzar la connexió amb el servidor. Aquesta finalització de la connexió utilitza els paquets FIN i ACK. Quan el servidor ja ha proporcionat les dades corresponents, aquest és el que inicia la finalització de manera que envia un segment amb el flag FIN a 1. El nostre equip doncs, haurà de correspondre amb un ACK i continuar fent el mateix que ha fet el servidor de manera que la connexió quedi totalment tancada.

Exercici 3

En aquest cas ocorre un comportament semblant, com que fem servir un domini en comptes d'una adreça IP, tornarem a utilitzar el protocol DNS per obtenir l'adreça corresponent i així poder connectar-nos al servidor que ens interessa. Quan ens arriba la resposta comencem a comunicar-nos amb el servidor en qüestió. No obstant però, la comunicació es portarà a terme fent servir un nou protocol, l'ICMP, el qual és un protocol de control de flux.

El protocol ICMP (Internet Communication Message Protocol) té com a finalitat el diagnòstic i el control d'una connexió, així com l'enviament d'un paquet d'error al servidor origen.

Tot i que la comanda ping fa servir l'ICMP com a protocol de comunicació entre host i servidor, no se li sol donar aquesta tasca.

```
C:\Users\Yuste32>ping www.google.com

Haciendo ping a www.google.com [216.58.201.164] con 32 bytes de datos:
Respuesta desde 216.58.201.164: bytes=32 tiempo=18ms TTL=51
Respuesta desde 216.58.201.164: bytes=32 tiempo=20ms TTL=51
Respuesta desde 216.58.201.164: bytes=32 tiempo=25ms TTL=51
Respuesta desde 216.58.201.164: bytes=32 tiempo=29ms TTL=51

Estadísticas de ping para 216.58.201.164:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 18ms, Máximo = 29ms, Media = 23ms
```

388	117.133029	192.168.1.19	192.168.1.1	DNS	74 Standard query 0xd0bf A www.google.com
389	117.134531	192.168.1.1	192.168.1.19	DNS	90 Standard query response 0xd0bf A www.google.com A 216.58.201.164
390	117.145355	192.168.1.19	216.58.201.164	ICMP	74 Echo (ping) request id=0x0001, seq=1044/5124, ttl=128 (reply in 391)
391	117.163472	216.58.201.164	192.168.1.19	ICMP	74 Echo (ping) reply id=0x0001, seq=1044/5124, ttl=51 (request in 390)
392	118.152737	192.168.1.19	216.58.201.164	ICMP	74 Echo (ping) request id=0x0001, seq=1045/5380, ttl=128 (reply in 393)
393	118.173276	216.58.201.164	192.168.1.19	ICMP	74 Echo (ping) reply id=0x0001, seq=1045/5380, ttl=51 (request in 392)
394	119.158781	192.168.1.19	216.58.201.164	ICMP	74 Echo (ping) request id=0x0001, seq=1046/5636, ttl=128 (reply in 395)
395	119.184584	216.58.201.164	192.168.1.19	ICMP	74 Echo (ping) reply id=0x0001, seq=1046/5636, ttl=51 (request in 394)
397	120.168683	192.168.1.19	216.58.201.164	ICMP	74 Echo (ping) request id=0x0001, seq=1047/5892, ttl=128 (reply in 398)
398	120.198414	216.58.201.164	192.168.1.19	ICMP	74 Echo (ping) reply id=0x0001, seq=1047/5892, ttl=51 (request in 397)

Figura 11. Paquets generats en executar ping

Quan ens disposem a analitzar el protocol ICMP ens trobem amb la següent informació:

```
▾ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0x4947 [correct]
    [Checksum Status: Good]
    Identifier (BE): 1 (0x0001)
    Identifier (LE): 256 (0x0100)
    Sequence number (BE): 1044 (0x0414)
    Sequence number (LE): 5124 (0x1404)
    [Response frame: 391]
  ▾ Data (32 bytes)
      Data: 6162636465666768696a6b6c6d6e6f707172737475767761…
      [Length: 32]
```

Figura 12. Informació del paquet ICMP

1) Type: 1 byte que representa si és una petició (value = 8) o una resposta (value = 0).

2) Code: 1 byte que pels ICMP de petició i resposta sempre serà 0.

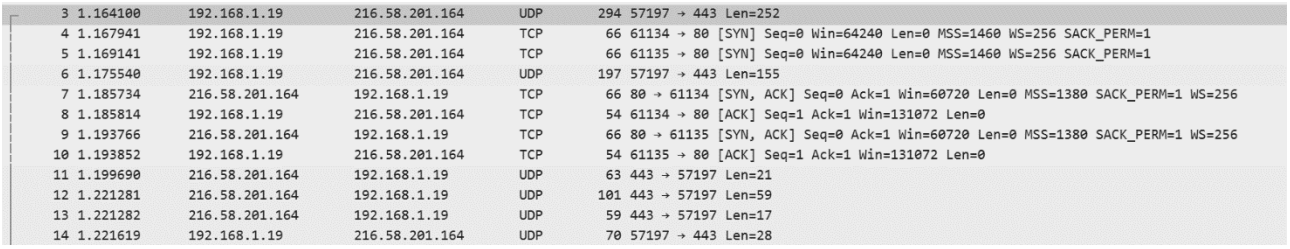
3) Identifier (BE)/(LE): identificador de 2 bytes que farem servir per comprovar que és la resposta que estem esperant (BE = big-endian, LE = little-endian).

4) Sequence number (BE)/(LE): té la mateixa finalitat que l'identifier, ens servirà per comprovar si és la nostra resposta.

5) Data: 32 bytes on anirà tota la informació.

6) Per últim, tant l'identificador, el nombre de seqüència i el data de la resposta ha de concordar amb la de la petició.

Si agafem la IP que ens ha retornat el protocol DNS, i l'introduïm el cercador d'un navegador web ens portarà directament a la pàgina principal de google amb la novetat que no tornarem a cridar un DNS (ja que l'adreça ja ens és coneguda).



3	1.164100	192.168.1.19	216.58.201.164	UDP	294	57197 → 443 Len=252
4	1.167941	192.168.1.19	216.58.201.164	TCP	66	61134 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
5	1.169141	192.168.1.19	216.58.201.164	TCP	66	61135 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
6	1.175540	192.168.1.19	216.58.201.164	UDP	197	57197 → 443 Len=155
7	1.185734	216.58.201.164	192.168.1.19	TCP	66	80 → 61134 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PERM=1 WS=256
8	1.185814	192.168.1.19	216.58.201.164	TCP	54	61134 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
9	1.193766	216.58.201.164	192.168.1.19	TCP	66	80 → 61135 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PERM=1 WS=256
10	1.193852	192.168.1.19	216.58.201.164	TCP	54	61135 → 80 [ACK] Seq=1 Ack=1 Win=131072 Len=0
11	1.199690	216.58.201.164	192.168.1.19	UDP	63	443 → 57197 Len=21
12	1.221281	216.58.201.164	192.168.1.19	UDP	101	443 → 57197 Len=59
13	1.221282	216.58.201.164	192.168.1.19	UDP	59	443 → 57197 Len=17
14	1.221619	192.168.1.19	216.58.201.164	UDP	70	57197 → 443 Len=28

Figura 13. Comunicació amb la ip de www.google.com

El sniffer captura la comunicació i l'intercanvi d'informació és que produeix entre el host (nosaltres) i el servidor (google), aquesta comunicació es fa mitjançant protocols UDP i TCP de manera alternada. Com hem vist abans utilitzarem TCP per a establir i finalitzar la connexió. Per a altres paquets es farà servir UDP.

En el protocol TCP s'ha fet servir el port 80 el qual és el port que s'escull per defecte a l'hora de realitzar una transferència d'[2]hipertext.
Com a control de flux podem veure que el protocol TCP fa ús dels flags SYN i ACK, el primer serveix per sincronitzar els números inicials de la seqüència, d'altra banda, el segon es retorna com a confirmació. Si ens fixem és el mateix que hem explicat amb anterioritat a l'exercici 2.

1) Linea 4/5: host envia TCP amb flag de sincronització activat

2) Linea 7/9: server retorna TCP amb flag de sincronització i confirmació activats

3) Linea 8/10: host retorna TCP amb flag de confirmació activat

[2] Sistema d'organització de la informació basat en la possibilitat de moure's per dins d'un text i cap a textos diferents per mitjà de paraules clau.

Exercici 4

Tot seguint les instruccions de l'enunciat, la xarxa muntada ens quedarà amb una estructura d'aquesta forma.

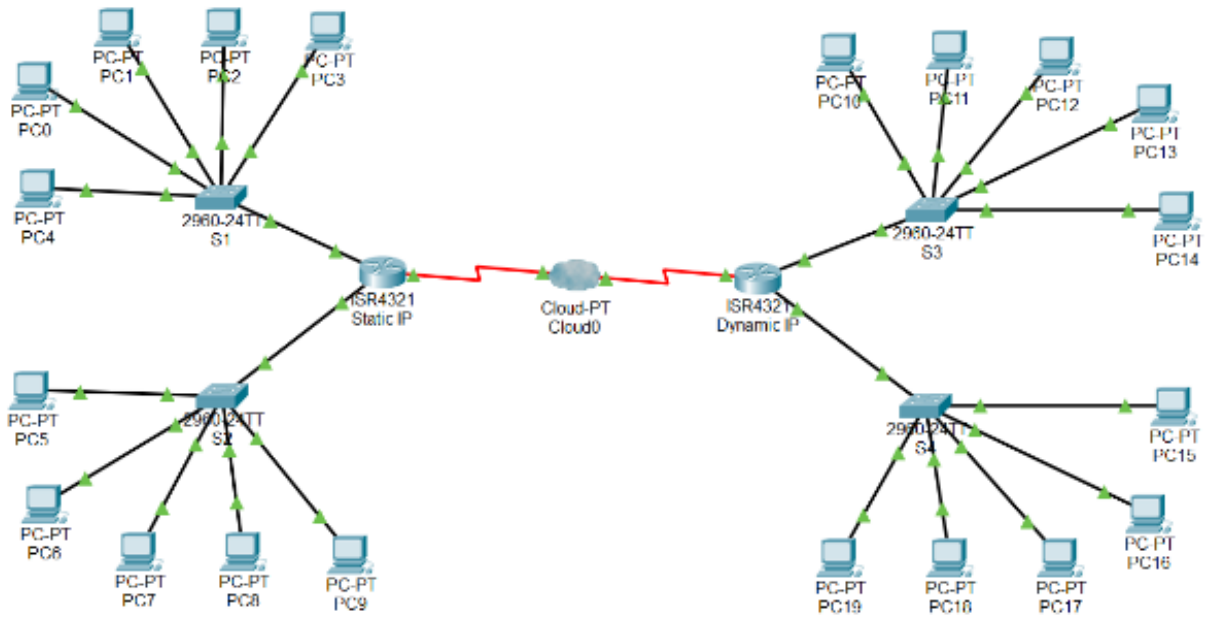


Figura 14. Primera xarxa muntada

Els equips de la xarxa de l'esquerra se'ls ha assignat una adreça IP manualment, però, els equips de la dreta han rebut la seva IP de manera dinàmica. Per aconseguir això hem hagut de configurar el router corresponent en mode DHCP de la següent manera.

1) ip dhcp pool "nom xarxa". Comanda amb la que entraren en el mode de configuració d'un nou servidor DHCP.

2) network "ip base" "mascara". Comanda per fixar el rang d'adreces a distribuir.

3) default-router "ip". Adreça que servirà de getaway pels equips (i que a més identificarà al router).

4) dns-server "ip". En cas que hi hagi un servidor DNS a la xarxa perquè l'assigni també a tots els dispositius.

5) I end per sortir del mode de configuració del servidor DHCP.

Per últim, ens hem d'assegurar que el router no assigna IP's reservades per tant les excloem.

6) ip dhcp excluded-address 192.168.1.1. IP d'exemple (correspon al router i per tant no la volem assignar a cap altre dispositiu).

En aquest moment, tots els ordinadors tenen una IP privada, però aquesta no ens permet navegar per internet, per tant hem de transformar-la a una pública si volem sortir de la nostra xarxa. És a dir, hem de configurar la NAT en els dos routers.

Primer assignarem una IP publica a la sortida del router i a continuació, habilitarem el port.

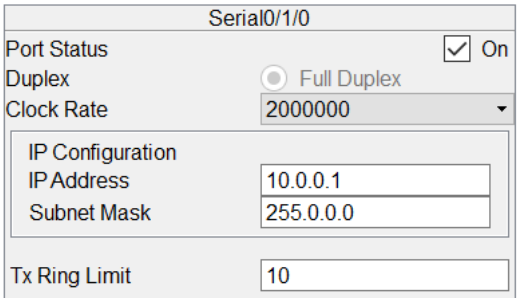


Figura 15. Configuració del router

Un cop fet això ens disposem a configurar-ho des de la consola un altre cop.

1) configure terminal. Mode configuració router.

2) acess-list 1 permit "ip" "mascara invertida". Transformarem totes les IP's contingudes en la introduïda a una IP pública.

3) ip nat inside source list 1 interface "port" indiquem per quin port sortirà l'adreça pública.

Per últim toca configurar els ports del router.

4) interface "nom del port interior". Entrem en mode configuració d'aquesta interfície.

5) ip nat inside. L'assignem com a port d'entrada.

6) exit. Sortim de la configuració.

7) intercace "nom del port de sortida". Tornem a entrar al mode de configuració de la interfície.

8) ip nat outside. Aquest cop el configurem com a port de sortida.

Ara només caldrà comprovar que funciona correctament.

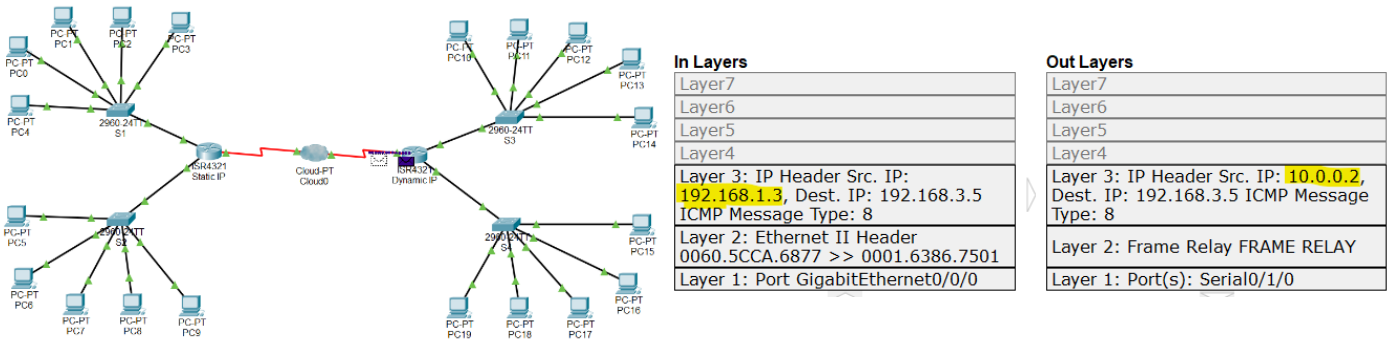


Figura 16. Transformació de la IP

Quan fem ping, es pot veure que l'adreça d'origen del paquet en arribar al router és diferent de l'adreça d'origen d'aquests mateix paquet en sortir. Per tant, podem deduïr que la xarxa s'ha configurat correctament.

Ara procedirem a la configuració d'una web.

Primer de tot necessitarem un servidor DNS que ens tradueixi el domini de la pàgina web a una adreça IP vàlida.

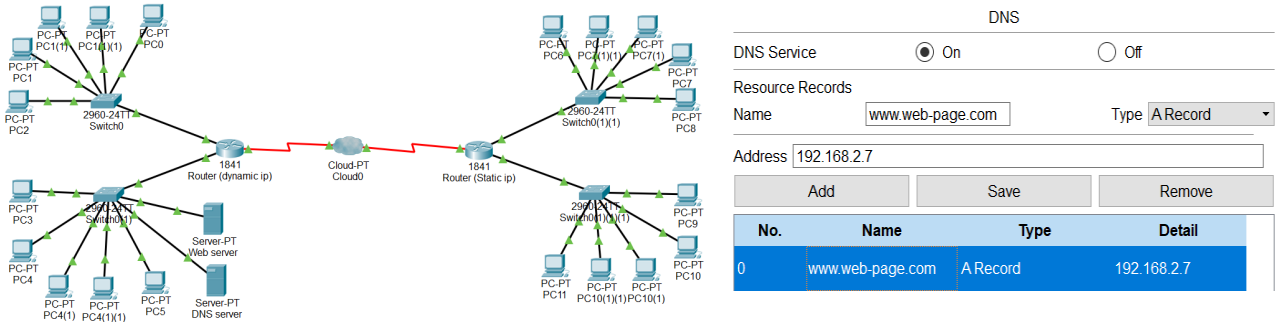


Figura 17. Introduïm el nom del domini amb la seva direcció IP corresponent

És important tenir en compte que tots els equips de la xarxa han de tenir l'adreça d'aquest servidor DNS per poder fer servir els dominis corresponents.

Un cop fet això, ja en podrem connectar a la pàgina web des d'un equip i quan ho fem es generen els següents paquets.

1.386	PC0	Switch0		DNS
1.387	Switch0	Router (dynamic ip)		DNS
1.388	Router (d…	Switch0(1)		DNS
1.389	Switch0(1)	DNS server		DNS
1.390	DNS server	Switch0(1)		DNS
1.391	Switch0(1)	Router (dynamic ip)		DNS

Figura 18. Part del recorregut del paquet *DNS*

L'equip que es vol connectar a la web envia un paquet DNS per tal que el servidor li respongui amb l'adreça de la web.

Quan el servidor retorna aquest paquet comença la comunicació entre la web i l'equip fent servir TCP i HTTP.

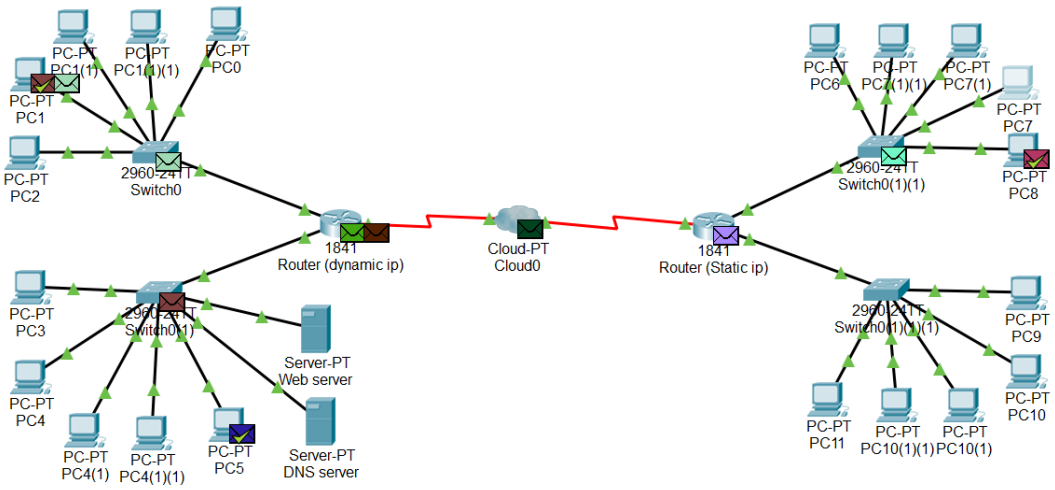
0.016	Switch0	PC0	TCP
0.016	--	PC0	HTTP
0.017	PC0	Switch0	TCP
0.017	--	PC0	HTTP
0.018	PC0	Switch0	HTTP
0.018	Switch0	Router (dynamic ip)	TCP
0.019	Switch0	Router (dynamic ip)	HTTP
0.019	Router (d…	Switch0(1)	TCP

Figura 19. Part de la comunicació entre equip i servidor web

I finalment, com a resultat se'ns mostraria la pàgina programada.

Figura 20. Pàgina web

Hem generat diversos paquets i peticions per veure com funciona la xarxa i el resultat és el següent.



0.009	Switch0	PC1	DNS
0.009	--	Router (dynamic ip)	ICMP
0.009	Switch0(1)	PC5	ICMP
0.009	Router (S…	Cloud0	ICMP
0.009	--	PC1	TCP
0.010	Router (d…	Switch0	ICMP
0.010	Router (d…	Cloud0	ICMP
0.010	Switch0(1)	Router (dynamic ip)	DNS
0.010	Switch0	PC1	TCP
0.010	Switch0(…	Router (Static ip)	ICMP
0.010	Router (S…	Cloud0	ICMP
0.010	PC1	Switch0	TCP

Figura 21. Visualització de la xarxa amb el tràfic generat

Conclusions

Generalment, hem après i estudiat com funcionen diferents protocols que fan servir la pila TCP/IP, com són el TCP, el DNS, el DAYTIME (en menor mesura) o l'ICMP, durant la realització dels diferents exercicis. Hem vist com s'encapsulen les dades per a ser transmeses entre dos terminals (més típicament hem vist l'exemple típic de l'arquitectura client-servidor, com ja vam fer a la pràctica anterior).

També hem vist com muntar una xarxa, fent servir més elements que en l'anterior pràctica així com configurar dues subxarxes diferents per a connectar-les entre si mitjançant Cisco Packet Tracer. També hem configurat el servidor DNS per a obrir una pàgina web a partir d'un nom de domini.

És interessant puntualitzar que tot això ho hem fet treballant amb un sniffer (Wireshark), que per a nosaltres és un software que no estem acostumats a utilitzar normalment i que hem trobat que és de molta utilitat, ja que ens serveix també per realitzar troubleshooting de la nostra xarxa personal en el cas que algun dia se'ns presentés un problema.

Podem dir doncs, que hem assolit amb èxit els subobjectius plantejats, així com l'objectiu general que hem vist a l'apartat Objectius de la pràctica.