

Noms:

Checa hernàndez, Martí & Rocha Guzmán, Alejandra Lisette

P1 - Sessió 2: Paràmetres i funcionament bàsic

Per a aquesta sessió de laboratori, continuarem treballant els conceptes de dispositius bàsics de xarxes de computadors i realitzarem algunes proves d'enviament Unicast i simularem Multicast.

Algunes comandes importants de Linux que s'utilitzaran a la primera sessió 1 de la pràctica 1:

- ip – llegiu-ne la documentació.
- scp – llegiu-ne la documentació. Si cal, instal·leu el paquet openssh-server.
- iperf – llegiu-ne la documentació.
- netcat – llegiu-ne la documentació.
- ping – llegiu-ne la documentació.

En una màquina Linux Ubuntu. Descarregueu-vos el vídeo "Warriors of the net" amb format H.264

(https://drive.google.com/file/d/1fFmhUektpsnUMfEdHWK5RkT77ITpnsdU/view?usp=drive_link o https://archive.org/details/warriors_of_net). En les dos màquines Ubuntu de treball, comproveu que teniu instal·lat, o instal·leu, el visualitzador de vídeo VLC (<https://www.videolan.org/>).



Figura 1. Icones del vídeo "Warriors of the Net" i el programa VLC.

Per configurar VLC per a transmetre per xarxa un flux de vídeo:

- Per fer proves es necessiten dos ordinadors a la mateixa xarxa.

Configuració del servidor:

- Obrir VLC, i premer Ctrl+s. Això obrirà la finestra Open Media.
- Seleccionar el vídeo a transmetre, i llavors botó Emission (o transmet) i després Next (o següent).
- Seleccionar new destination: HTTP i apretar Add. Llavors la adreça IP és la de l'ordinador (el VLC ja sap quina agafar) que estem configurant el VLC (el port 8080) i en path no cal posar res.
- Seleccionar transcoding: "Video - H.264 + MP3 (TS)"
- Posar la reproducció en mode continu.

El vídeo pot trigar una estona a fer l'streaming (o sabrem que està fent streaming perquè la barra de reproducció començarà a avançar).

Configuració del client

- En el VLC. Prémer Ctrl+n.
- http://ipaddr:source:8080 (per exemple http://192.168.61.168:8080)
- Llavors prémer play (es pot prémer bucle infinit perquè es reproduïx de forma continua).

Així doncs, a continuació es realitzaran diferents proves de modes de transmissió (Unicast, Broadcast, Multicast...) alhora que es comprovaran les diferents velocitats de transmissió a la xarxa LAN del laboratori.

1. Copieu el fitxer de vídeo que heu descarregat en una màquina a la carpeta de Baixades (home/pac/username) i també envieu a la carpeta de Baixades de la màquina del teu/va company/a de laboratori (utilitzant la comanda scp). Les màquines primer hauran de tenir instal·lat openssh-server (sudo apt install openssh-server).

```
e9501366@aul-1614:~/Baixades$ scp trailer.mp4 f5160909@192.168.60.22:/home/est/f5160909/Baixades
f5160909@192.168.60.22's password:
trailer.mp4                                100% 5729KB 107.6MB/s   00:00

f5160909@aul-1604:~/Baixades$ ls
trailer.mp4
```

2. Quin mode de transmissió hem realitzat (Unicast, Broadcast, o Multicast)? Raona el perquè. Quina velocitat de transmissió hem obtingut?

El mètode de transmissió que hem utilitzat és Unicast ja que hem especificat a nivell singular a quina màquina volíem transmetre dades. Hem seleccionat de tots els ordinadors de l'aula un ordinador singular, i aquest ha sigut l'únic que ha rebut la informació. Hem seleccionat 1 possible destí. La velocitat de transmissió és de 107.6MB/s que és bastant inferior a la de la comanda iperf ja que a l'hora de transmetre dades a través del servei SSH hi ha un xifratge dels arxius (una codificació de font i canal) que enralenteix la transmissió de dades

Amb el Wireshark activat a l'interfície corresponent d'un dels host, comproveu quin és l'ample de banda disponible a la LAN de laboratori, com s'explica en els següents apartats.

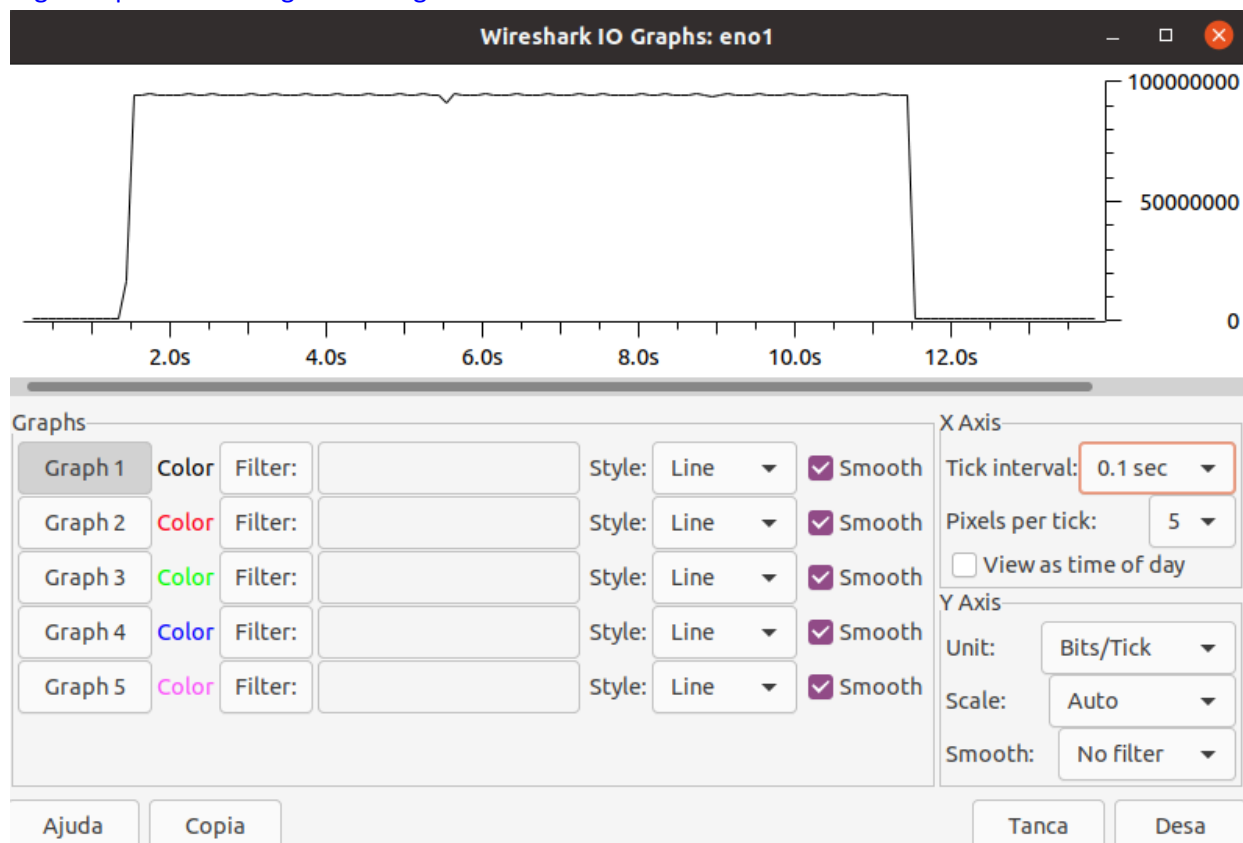
3. Genera tràfic amb l'eina *iperf* durant 15s, utilitzant PC1 com a servidor i PC2 com a client. Quin és el temps de transmissió? Quin és l'ample de banda de la connexió? Mostreu la gràfica amb el Wireshark (IO Graph). Es correspon amb el valor de velocitat de transmissió obtingut al punt 2?

```
f5160909@aul-1604:~/Baixades$ iperf -s
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
[  4] local 192.168.60.22 port 5001 connected with 192.168.60.128 port 54314
[ ID] Interval      Transfer    Bandwidth
[  4] 0.0-10.0 sec  1.09 GBytes  934 Mbits/sec

e9501366@aul-1614:~/Baixades$ iperf -c 192.168.60.22
-----
Client connecting to 192.168.60.22, TCP port 5001
TCP window size: 876 KByte (default)
-----
[  3] local 192.168.60.128 port 54314 connected with 192.168.60.22 port 5001
[ ID] Interval      Transfer    Bandwidth
[  3] 0.0-10.0 sec  1.09 GBytes  935 Mbits/sec
```

Suposant que la velocitat de transmissió (o Bandwidth) de 935×10^6 bits/sec ha sigut més o menys constant durant l'enviament de dades, enviar un arxiu de 1.09 GBytes (que són $1.09 \times 10^9 \times 8 = 8720000000$ Bits) haurà trigat $8720000000 / 935000000 = 9.326$ segons.

El gràfic que hem obtingut és el següent:



Podem observar que més o menys la nostra suposició de que la transferència ha durat 9.326 segons és el que es mostra al gràfic. A més, si volguéssim assegurar-nos, podríem utilitzar la comanda de "time" d'UNIX per que ens calculi quant de temps ha trigat en executar la comanda desde el client, que més o menys acaba amb el mateix valor.

```
e9501366@aui-1614:~/Baixades$ time iperf -c 192.168.60.22
-----
Client connecting to 192.168.60.22, TCP port 5001
TCP window size: 1.00 MByte (default)
-----
[ 3] local 192.168.60.128 port 49746 connected with 192.168.60.22 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3]  0.0-10.0 sec  1.09 GBytes   936 Mbits/sec

real    0m10.058s
user    0m0.025s
sys     0m0.455s
```

- Configureu *PC2* per a transmetre per xarxa un flux de vídeo amb el programa VLC, tal i com s'explica a l'inici del Pla de treball. Reproduïu aquest vídeo al *PC1* mentre captureu el tràfic també a *PC1* amb Wireshark (filtre: ip.addr==X.X.X.X_servidor and tcp.port==8080).

*eno1 [Wireshark 1.12.13 (v1.12.13-0-g969649d from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Desa

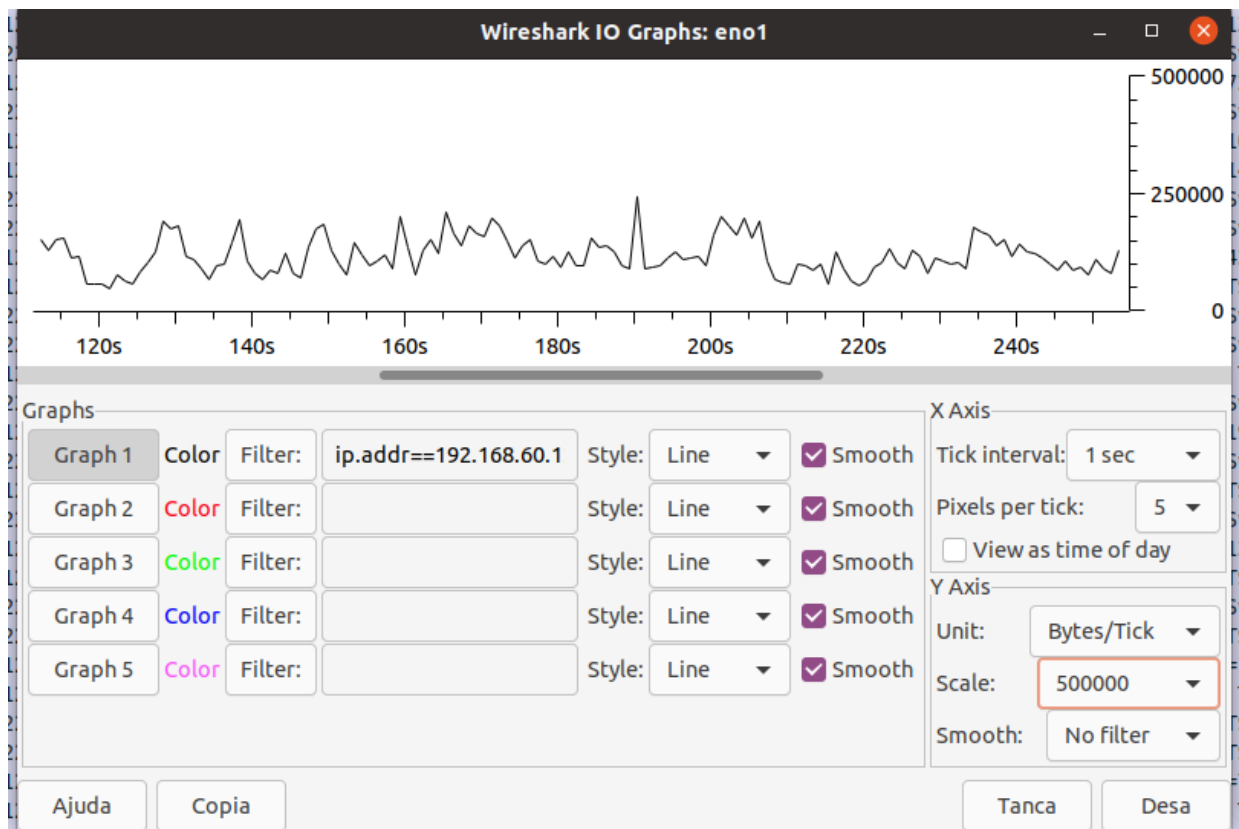
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.60.128	192.168.60.22	TCP	10066	8080->55788 [PSH, ACK] Seq=1 Ack=1 Win=509 Len=10000 TSval=159012064 TSecr=3540585833
2	0.000003000	192.168.60.128	192.168.60.22	TCP	5858	8080->55788 [ACK] Seq=10001 Ack=1 Win=509 Len=5792 TSval=159012064 TSecr=3540585833
3	0.000031000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=10001 Win=24523 Len=0 TSval=3540586094 TSecr=159012064
4	0.000048000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=15793 Win=24493 Len=0 TSval=3540586094 TSecr=159012064
5	0.000103000	192.168.60.128	192.168.60.22	TCP	13098	8080->55788 [PSH, ACK] Seq=15793 Ack=1 Win=509 Len=13032 TSval=159012064 TSecr=3540585833
6	0.000110000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=28825 Win=24424 Len=0 TSval=3540586094 TSecr=159012064
7	0.000376000	192.168.60.128	192.168.60.22	TCP	7338	8080->55788 [PSH, ACK] Seq=28825 Ack=1 Win=509 Len=7272 TSval=159012064 TSecr=3540586094
8	0.000388000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=36097 Win=24535 Len=0 TSval=3540586094 TSecr=159012064
10	0.241523000	192.168.60.128	192.168.60.22	TCP	10066	8080->55788 [PSH, ACK] Seq=36097 Ack=1 Win=509 Len=10000 TSval=159012305 TSecr=3540586094
11	0.241528000	192.168.60.128	192.168.60.22	TCP	14546	8080->55788 [PSH, ACK] Seq=46097 Ack=1 Win=509 Len=14480 TSval=159012305 TSecr=3540586094
12	0.241550000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=46097 Win=24523 Len=0 TSval=3540586335 TSecr=159012305
13	0.241564000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=60577 Win=24448 Len=0 TSval=3540586335 TSecr=159012305
14	0.241578000	192.168.60.128	192.168.60.22	TCP	4410	8080->55788 [PSH, ACK] Seq=60577 Ack=1 Win=509 Len=4344 TSval=159012305 TSecr=3540586094
15	0.241579000	192.168.60.128	192.168.60.22	TCP	1514	8080->55788 [ACK] Seq=64921 Ack=1 Win=509 Len=1448 TSval=159012305 TSecr=3540586094
16	0.241583000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=64921 Win=24424 Len=0 TSval=3540586335 TSecr=159012305
17	0.241587000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=66369 Win=24417 Len=0 TSval=3540586335 TSecr=159012305
18	0.241874000	192.168.60.128	192.168.60.22	TCP	17442	8080->55788 [ACK] Seq=66369 Ack=1 Win=509 Len=17376 TSval=159012306 TSecr=3540586335
19	0.241887000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=83745 Win=24485 Len=0 TSval=3540586336 TSecr=159012306
20	0.242159000	192.168.60.128	192.168.60.22	TCP	2050	8080->55788 [PSH, ACK] Seq=83745 Ack=1 Win=509 Len=1984 TSval=159012306 TSecr=3540586335
21	0.242174000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=85729 Win=24560 Len=0 TSval=3540586336 TSecr=159012306
47	0.484413000	192.168.60.128	192.168.60.22	TCP	8754	8080->55788 [ACK] Seq=85729 Ack=1 Win=509 Len=8688 TSval=159012548 TSecr=3540586336
48	0.484432000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=94417 Win=24530 Len=0 TSval=3540586578 TSecr=159012548
49	0.484497000	192.168.60.128	192.168.60.22	TCP	1378	8080->55788 [PSH, ACK] Seq=94417 Ack=1 Win=509 Len=1312 TSval=159012548 TSecr=3540586336
50	0.484501000	192.168.60.128	192.168.60.22	TCP	8754	8080->55788 [ACK] Seq=95729 Ack=1 Win=509 Len=8688 TSval=159012548 TSecr=3540586336
51	0.484520000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=95729 Win=24523 Len=0 TSval=3540586578 TSecr=159012548
52	0.484545000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=104417 Win=24456 Len=0 TSval=3540586578 TSecr=159012548
53	0.484631000	192.168.60.128	192.168.60.22	TCP	10282	8080->55788 [PSH, ACK] Seq=104417 Ack=1 Win=509 Len=10136 TSval=159012548 TSecr=3540586336
54	0.484634000	192.168.60.128	192.168.60.22	TCP	4410	8080->55788 [ACK] Seq=114553 Ack=1 Win=509 Len=4344 TSval=159012548 TSecr=3540586336
55	0.484653000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=114553 Win=24424 Len=0 TSval=3540586578 TSecr=159012548
56	0.484666000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=118897 Win=24477 Len=0 TSval=3540586578 TSecr=159012548
57	0.484843000	192.168.60.128	192.168.60.22	TCP	7306	8080->55788 [PSH, ACK] Seq=118897 Ack=1 Win=509 Len=7240 TSval=159012548 TSecr=3540586336
58	0.484846000	192.168.60.128	192.168.60.22	TCP	8754	8080->55788 [ACK] Seq=126137 Ack=1 Win=509 Len=8688 TSval=159012549 TSecr=3540586578
59	0.484854000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=126137 Win=24538 Len=0 TSval=3540586579 TSecr=159012548
60	0.484861000	192.168.60.128	192.168.60.22	TCP	66	55788->8080 [ACK] Seq=1 Ack=134825 Win=24493 Len=0 TSval=3540586579 TSecr=159012549
61	0.485098000	192.168.60.128	192.168.60.22	TCP	2962	8080->55788 [PSH, ACK] Seq=134825 Ack=1 Win=509 Len=2896 TSval=159012549 TSecr=3540586578
62	0.485091000	192.168.60.128	192.168.60.22	TCP	150	8080->55788 [PSH, ACK] Seq=137721 Ack=1 Win=509 Len=84 TSval=159012549 TSecr=3540586578

Transmission Control Protocol, Src Port: 8080 (8080), Dst Port: 55788 (55788), Seq: 1, Ack: 1, Len: 10000

0000 50 65 f3 46 75 64 50 65 f3 50 8d fe 08 00 45 00 Pe.FudPe .P....E.
0010 27 44 3c 14 40 00 08 06 dd b8 c0 a8 3c 80 c0 a8 'D<.@.
0020 3c 16 1f 90 d9 c9 e9 b2 35 16 fa 00 2e c9 08 18 <..... 5.....
0030 01 fd 21 1e 00 00 01 01 08 09 09 7a 54 e0 d3 092T....
0040 0d 69 47 40 00 37 a6 00 ff ff ff ff ff ff ff ff ..100.7.....

File: "/tmp/wireshark_pcapng..." Packets: 42783 · Displayed: 16766 (39,2%) · Dropped: 0 (0,0%) Profile: Default

5. Quina velocitat de transmissió requereix aquesta flux de vídeo (Statistics->IO graph de wireshark)? Quin mode de transmissió simula (Unicast, Broadcast, o Multicast)?



Mirant el gràfic, havent-hi filtrant els paquets capturats només per els que ens hem estant enviant a través del fluxe de video entre el client i servidor, podem assumir que la transferència de dades no superava els 250 KBytes per segon on transmet informació. Per evitar bottlenecks, podem assumir que la velocitat de transferència hauria de ser superior a aquest valor, encara que la mitjana no superava els 150~ KBytes.

El mètode de transferència que simula és Multicast. Podem deduir això de la manera que el nostre servidor (192.168.60.128) està enviant paquets a la xarxa. Podem descartar que estigui enviant-los en Broadcast ja que en cap moment està enviant informació a la IP de Broadcast. Tambè podem descartar que només estigui enviant a un destí ja que està intentant enviar informació a diferents equips. Per tant, la seva transmissió s'està fent en Multicast, on el nostre ordinador client és l'únic destí que està aprofitant aquesta informació.

6. Tindrem problemes en aquest escenari per reproduir el streaming de vídeo a *PC1*? I quan estiguin els diferents ordinadors de laboratori demanant de connectar-se al vídeo simultàniament? Fins quants ordinadors suportaria la LAN del laboratori demanant el vídeo? Raoneu la resposta i feu càlculs pertinents per argumentar-ho.

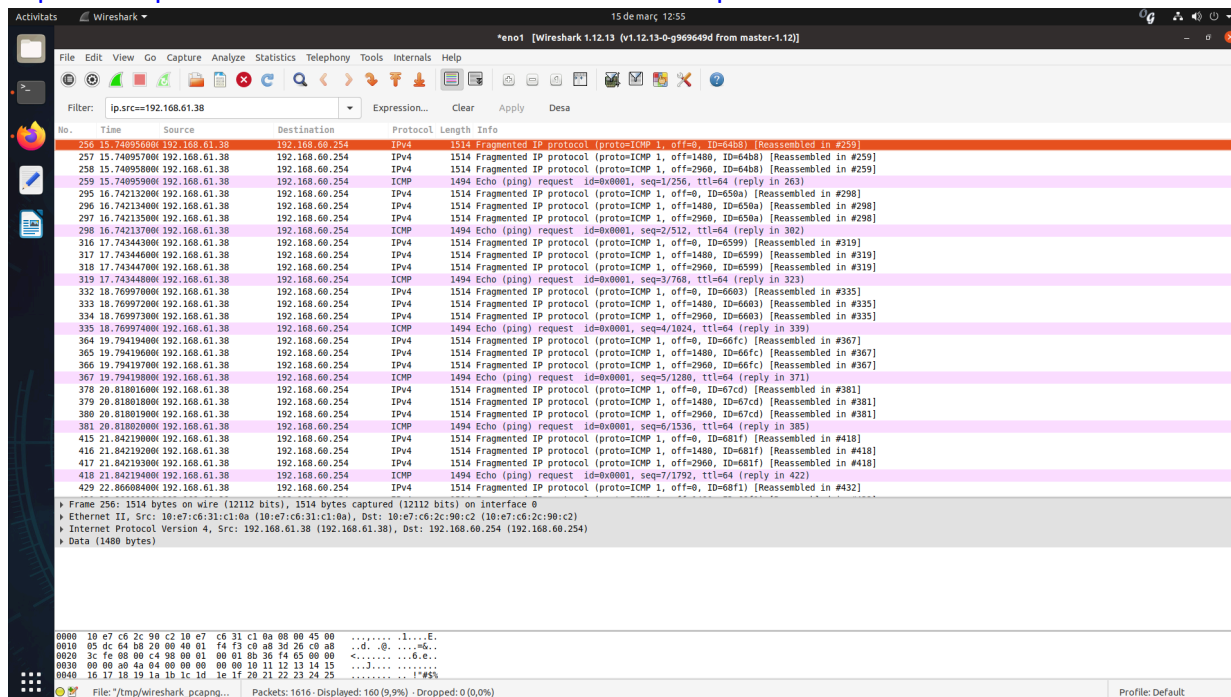
La quantitat d'ordinadors que podria suportar la xarxa LAN dependrà de l'enllaç que utilitzem per connectar a tots els dispositius. Suposant que estem utilitzant cable de CAT5 (1 Gbps) per connectar tots els dispositius, que la nostra NIC pot enviar fins a 1 Gbps, que la velocitat de transferència promitja sigui de 250 KBytes (o $250 * 1024 = 256000$ Bits) per segon i que la fórmula de la Velocitat de Transmissió sigui $V_T = \text{Size} / \text{Speed}$, podem enviar-li informació fins a $1 * 10^9 / 250 * 10^3 = 4000$ ordinadors. Això només es donarà en una situació ideal, i si no hi ha cap problema. A més, no estem tenint en compte si la targeta NIC del nostre dispositiu (o procesador, ja que s'ha de tindre en compte el codificador de canal font) pot suportar aquestes velocitats.

Continuant amb l'estudi de dispositius lògics de xarxes de comunicació, és important conèixer les particularitats dels dispositius bàsics per poder construir xarxes LAN comunicades amb xarxes WAN de forma efectiva.

7. Utilitzant la comanda ping, genereu tràfic interrogant a un altre ordinador, enviant-li 5892 Bytes. Mitjançant Wireshark, incloïu la captura i indiqueu el filtre que heu utilitzat. Quan fragments s'han creat? Raoneu el perquè. Quines són els tamanys de capçaleres dels protocols involucrats (Ethernet, IP i ICMP)? Calculeu l'eficiència d'encapsulament IP per a aquests paquets.

```
e9501366@aul-1916:~$ ping 192.168.60.254 -s 5892
PING 192.168.60.254 (192.168.60.254) 5892(5920) bytes of data.
5900 bytes from 192.168.60.254: icmp_seq=1 ttl=64 time=0.953 ms
5900 bytes from 192.168.60.254: icmp_seq=2 ttl=64 time=0.792 ms
5900 bytes from 192.168.60.254: icmp_seq=3 ttl=64 time=0.778 ms
5900 bytes from 192.168.60.254: icmp_seq=4 ttl=64 time=0.718 ms
5900 bytes from 192.168.60.254: icmp_seq=5 ttl=64 time=0.782 ms
5900 bytes from 192.168.60.254: icmp_seq=6 ttl=64 time=0.457 ms
5900 bytes from 192.168.60.254: icmp_seq=7 ttl=64 time=0.736 ms
5900 bytes from 192.168.60.254: icmp_seq=8 ttl=64 time=0.638 ms
5900 bytes from 192.168.60.254: icmp_seq=9 ttl=64 time=0.799 ms
5900 bytes from 192.168.60.254: icmp_seq=10 ttl=64 time=0.796 ms
```

Captura de pantalla del Wireshark i el filtre utilitzat és ip.src==192.168.61.28:



Com es mostra a Wireshark, cada “ping” que hem fet de l’ordinador font (192.168.61.38) a l’ordinador destí (192.168.60.254) s’ha separat en 3 fragments diferents. Podem comprovar que aquests fragments van un rere l’altre ja que el seu ID únic és igual entre cada tupla, i el seu offset – que és el següent paquet on es va tallar el paquet previ – es talla de manera correcta. Encara que un paquet ping, que segueix el protocol IP, tingui un màxim de 65535 bytes que és un tamany molt superior al dels nostres pings, es fragmenta a nivell de xarxa per poder-ho enviar més eficientment. Per això el paquet ICMP que és el que crida la comanda ping com a tal ja que és un servei de control surt sencer, mentres que la informació com a tal s’ha fragmentat.

Els headers dels paquets IP són de 20 bytes, els headers dels paquets ICMP són de 8 bytes, i la trama ethernet són 14 bytes. Aquestes dades les hem obtingut de la informació que proporciona Wireshark, restant el tamany de les dades del paquet amb els diferents nivells d’encapsulament per obtenir quants bytes de header han afegit.

La seva eficiència d’encapsulament la podem comprovar si calculem quants bytes de dades tindrem en cada paquet per a cada byte de header que encapsulem. En el cas dels paquets IP amb un header the 20 bytes, l’eficiència de l’encapsulament serà de $(1480/1500) * 100 = 98.667\%$

8. Llavors, seleccionant un filtre a Wireshark només de TCP (filtre tcp), poseu a capturar amb Wireshark, i navegueu una estona per Internet. Pareu la captura i ordeneu els paquets pel camp Length (de més gran a més petit). Apareixen paquets amb una longitud més gran que la MTU? Com és que s'han pogut transmetre paquets més grans que la MTU? Calculeu la eficiència d'encapsulament IP per als paquets més petits i més grans que hagueu trobat a la captura, i comenteu-ne els valors.

Captura del Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
32920	69.18512000	151.101.133.140	192.168.60.254	TLSv1.2	41466	Application Data
27681	63.16148600	151.101.133.140	192.168.60.254	TLSv1.2	41466	Application Data
29395	67.25637600	151.101.133.140	192.168.60.254	TLSv1.2	38706	Application Data
25192	42.50876500	151.101.133.140	192.168.60.254	TLSv1.2	38706	Application Data
28621	67.03629000	151.101.133.140	192.168.60.254	TLSv1.2	35946	Application Data
28608	67.03508600	151.101.133.140	192.168.60.254	TLSv1.2	35946	Application Data
28045	63.39058900	151.101.133.140	192.168.60.254	TLSv1.2	35946	Application Data
27958	63.34715400	151.101.133.140	192.168.60.254	TLSv1.2	35946	Application Data
27820	63.29079600	151.101.133.140	192.168.60.254	TLSv1.2	35946	Application Data
27661	63.20175300	151.101.133.140	192.168.60.254	TLSv1.2	35946	Application Data
25296	42.70834900	151.101.133.140	192.168.60.254	TLSv1.2	35946	Application Data
25098	42.46480300	151.101.133.140	192.168.60.254	TLSv1.2	35946	Application Data
25551	43.06436100	88.221.213.185	192.168.60.254	TLSv1.2	34818	Application Data
4544	27.22151700	92.123.32.167	192.168.60.254	TLSv1.2	34818	Application Data
39643	136.62574700	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
32872	69.15300700	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
29814	67.49761600	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
29718	67.44597400	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
29413	67.27694700	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
29384	67.25520000	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
28453	66.95851800	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
27944	63.31536300	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
27750	63.24091000	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
27385	63.08065600	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
27270	63.01192000	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
25248	42.54002500	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
25209	42.52200700	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
25160	42.49720400	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
25152	42.49665100	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
25138	42.49430400	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
25091	42.46359300	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
19799	40.65982100	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
10370	37.02010000	151.101.133.140	192.168.60.254	TLSv1.2	34566	Application Data
Frame 32920: 41466 bytes on wire (331728 bits), 41466 bytes captured (331728 bits) on interface 0						
Ethernet II, Src: Fortinet a7:b3:32 (00:09:0f:a7:b3:32), Dst: 10:e7:c6:2c:90:c2 (10:e7:c6:2c:90:c2)						
Internet Protocol Version 4, Src: 151.101.133.140 (151.101.133.140), Dst: 192.168.60.254 (192.168.60.254)						
Transmission Control Protocol, Src Port: 443 (443), Dst Port: 46882 (46882), Seq: 95052375, Ack: 59281, Len: 41400						
[3 Reassembled TCP Segments (16406 bytes): #32917(154), #32918(6900), #32920(9352)]						
Secure Sockets Layer						
Secure Sockets Layer						
0000 10 e7 c6 2c 90 c2 00 09 0f a7 b3 32 08 00 45 a4 2..E.						
0010 a1 ec 52 f1 40 00 37 06 33 de 97 65 85 8c c0 a8 ..R.@.7. 3..e....						
0020 3c fe 01 bb b7 22 14 ac 13 cb e5 c1 e4 dd 00 18 <.....						
Frame (41466 bytes) Reassembled TCP (16406 bytes)						
File: /tmp/wireshark_pcapng... Packets: 39927 - Displayed: 31649 (79,3%) - Dropped: 0 (0,0%)						

Tenint en compte que el MTU d'un paquet TCP/IP és de 65535 bytes, i que el paquet més gran que em capturat és de 41466 bytes, podem observar que no podem tindre paquets més grans al MTU. En el cas que un packet necessitesi enviar més informació (tenint en compte el tamany del header), el packet es fragmentaria com hem observat a l'exercisi previ. En el cas que un paquet necessitesi enviar un missatge superior al MTU, el missatge s'enviaria fragmentat per poder enviar tot el missatge. Aquesta fragmentació es pot evitar amb una flag de control, pero per defecte aquesta fragmentació es farà automàticament.

L'eficiència d'encapsulament IP depèn molt del tamany de les dades. Com que el tamany del header és invariable, l'única variable que canvia aquesta eficiència és la quantitat de dades útils que enviem. Per al cas més gran, la nostra eficiència seria de $(41466 / 41586) * 100 = 99.711\%$, mentres que el del paquet més petit serà de $(54/74) * 100 = 72.973\%$

Visita a les instal·lacions de Telecomunicació de la EPSEVG:

Aquest apartat el realitzareu i completareu durant la primera part de la Pràctica 2. Lleixiu bé les preguntes a continuació i comenceu-les a preparar per a la visita guiada. Durant la visita guiada haureu de fer les preguntes oportunes per completar adequadament les dos qüestions a continuació.

Per treballar les respostes a continuació, utilitzeu els plànols de l'Escola (podeu utilitzar-los com a referència) disponibles a:

<https://epsevg.upc.edu/ca/escola/espais>

9. Dibuixeu esquemàticament sobre un plànol de l'Escola que s'identifiquin les diferents sales d'interconnexió i el laboratori de l'assignatura. Dibuixeu, sobre el plànol de l'Escola, la topologia de la xarxa de l'Escola. Incloeu els dispositius bàsics de xarxa (indicant de quin tipus de dispositiu bàsic de xarxa es tracta i quina funció té dins de la xarxa). Almenys indiqueu on es troba el modem d'Internet, el router per defecte de l'escola i el switch de laboratori.
10. Pel que fa als mitjans de transmissió de dades que heu vist. Quin tipus de mitjà s'utilitza en el cablejat horitzontal? ¿Hi ha alguna restricció quant a la distància? Quin mitjà de transmissió s'utilitza per connectar els distribuïdors (cablejat vertical)? ¿Hi ha alguna restricció quant a la distància? Enumera els diferents trams de mitjà de transmissió i els diferents equips que hi ha entre una roseta de dades situada al laboratori i la sortida externa de l'Escola. Exemple: roseta de dades - tram de cable A (x metres) - sala B amb equips C i D - tram de cable E, etc.