



Universidad
Rey Juan Carlos

Escuela Técnica Superior
Ingeniería de Telecomunicación

TECNOLOGÍAS DE TELEVISIÓN EN INTERNET

PRÁCTICA 3: INTERACCIÓN DE APLICACIONES EN REDES IP

Yolanda Lillo Mata

DNI: -

Correo: y.lillo.2016@alumnos.urjc.es

Recordamos que para configurar la emisión de vídeo lo hacemos como en la práctica 2, aquí el servidor de video es el cliente iPerf y el cliente de vídeo es el servidor iPerf. Repasamos rápidamente como se hacía, para el servidor se configura en la opción medio, emitir, añadimos el video, el protocolo (UDP o RTP para esta práctica) y la ip del cliente así como su puerto. Para el cliente en la opción medio, añadir ubicación de red y completamos con la ip y con el mismo puerto que el servidor.

1. Emisión de vídeo en UDP + Aplicación UDP

- Indicad las características de la red que muestra la aplicación iPerf.

Para este apartado, he configurado iPerf del siguiente modo:

- Cliente: `iperf -c (IPSERVIDOR) -u -b 1G -i 1 -t 600`, donde -c significa que se trata de un cliente, la ip del servidor que en este caso es 212.128.254.47, -u para recibir datagramas UDP, -b para indicar el ancho de banda, -i son los segundo en los cuales se volverá a realizar la medición y -t que es el tiempo que va a durar nuestra transición, un tiempo amplio para que dé tiempo a recibir el vídeo.
- Servidor: `iperf -s -u -i 1`

Ambas máquinas conectadas al puerto 5001, el tamaño del buffer de 208 kByte que es el que viene por defecto y la ip del cliente es 212.128.255.71.

- Indicad el comportamiento de la aplicación iPerf antes de la emisión/recepción de vídeo. Vemos el comportamiento de la aplicación antes de iPerf en las siguientes figuras.

```

yllo@f-l3208-pc07:~$ iperf -c 212.128.254.47 -u -b 1G -i 1 -t 600
.....
Client connecting to 212.128.254.47, UDP port 5001
Sending 1470 byte datagrams, IPG target: 10.95 us (kalman adjust)
UDP buffer size: 208 KByte (default)
.....
[ 3] local 212.128.255.71 port 34915 connected with 212.128.254.47 port 5001
[ ID] Interval      Transfer      Bandwidth
[ 3] 0.0- 1.0 sec   113 MBytes   944 Mbits/sec
[ 3] 1.0- 2.0 sec   110 MBytes   921 Mbits/sec
[ 3] 2.0- 3.0 sec   113 MBytes   944 Mbits/sec
[ 3] 3.0- 4.0 sec   113 MBytes   946 Mbits/sec
[ 3] 4.0- 5.0 sec   113 MBytes   947 Mbits/sec
[ 3] 5.0- 6.0 sec   113 MBytes   952 Mbits/sec

```

Figura 1. Comportamiento cliente en la aplicación iPerf

```

yllo@f-l3109-pc07:~$ iperf -s -u -i 1
.....
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 208 KByte (default)
.....
[ 3] local 212.128.254.47 port 5001 connected with 212.128.255.71 port 34915
[ ID] Interval      Transfer      Bandwidth      Jitter      Lost/Total Datagram
[ 3] 0.0- 1.0 sec   113 MBytes   944 Mbits/sec   0.026 ms    0/80270 (0%)
[ 3] 1.0- 2.0 sec   110 MBytes   921 Mbits/sec   0.025 ms    0/78286 (0%)
[ 3] 2.0- 3.0 sec   113 MBytes   944 Mbits/sec   0.027 ms    66/80332 (0.082%)
[ 3] 3.0- 4.0 sec   113 MBytes   945 Mbits/sec   0.025 ms    36/80392 (0.045%)
[ 3] 4.0- 5.0 sec   113 MBytes   947 Mbits/sec   0.017 ms    4/80567 (0.005%)
[ 3] 5.0- 6.0 sec   114 MBytes   952 Mbits/sec   0.015 ms    0/80974 (0%)

```

Figura 2. Comportamiento servidor en la aplicación iPerf

- Indicad el comportamiento de la aplicación iPerf durante la emisión/recepción de vídeo. Observamos que el vídeo va mucho más lento, se para, tiene distorsiones y se ve como más difuminado.

- ¿El tráfico de vídeo se ve afectado por el tráfico de iPerf?

Observamos que aumenta la pérdida de paquetes en el servidor y disminuye el ancho de banda en el cliente.

- Adjunta la captura de Wireshark y describe brevemente los paquetes pertenecientes al flujo de la comunicación que se ha desarrollado entre los equipos, mencionando los protocolos que toman partido en la comunicación y los pasos que se llevan a cabo mediante los mensajes de los protocolos.

35...	53.522151...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522153...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522155...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522336...	193.147.79.63	212.128.255.71	TCP	82 58846 → 8080	[PSH, ACK] Seq=66073 Ack=19981510 Win=23569 Len=16 TSval=1055
35...	53.522392...	212.128.255.71	193.147.79.63	TCP	60 8080 → 58846	[ACK] Seq=19981510 Ack=66089 Win=63 Len=0 TSval=543661155
35...	53.522788...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522794...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522796...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522798...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522800...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522802...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522804...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522807...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522809...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522811...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522813...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522815...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522817...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
Frame 1: 202 bytes on wire (1616 bits), 202 bytes captured (1616 bits) on interface enp0s31f6, id 0						
Ethernet II, Src: Dell_cb:65:dd (48:4d:7e:cb:65:dd), Dst: JuniperN_f1:e7:f0 (7c:e2:ca:f1:e7:f0)						
Internet Protocol Version 4, Src: 212.128.255.71, Dst: 10.110.96.11						
Transmission Control Protocol, Src Port: 816, Dst Port: 2049, Seq: 1, Ack: 1, Len: 136						
Remote Procedure Call, Type: Call, XID: 0xa5a05cb3						
35...	53.521361...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.521599...	212.128.255.71	212.128.254.47	MPEG TS	1358	[MP2T Fragment of a reassembled packet]
35...	53.522008...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522027...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522029...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522031...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522033...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522035...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522040...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522042...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522044...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522046...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522048...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522050...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522052...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522054...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522056...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
35...	53.522059...	212.128.255.71	212.128.254.47	UDP	1512 34915 → 5001	Len=1470
Frame 3559425: 1358 bytes on wire (10864 bits), 1358 bytes captured (10864 bits) on interface enp0s31f6, id 0						
Ethernet II, Src: Dell_cb:65:dd (48:4d:7e:cb:65:dd), Dst: Dell_f9:5a:11 (e4:b9:7a:f9:5a:11)						
Internet Protocol Version 4, Src: 212.128.255.71, Dst: 212.128.254.47						
User Datagram Protocol, Src Port: 60990, Dst Port: 5001						
ISO/IEC 13818-1 PID=0x64 CC=9						
Reassembled in: 3561173						
ISO/IEC 13818-1 PID=0x64 CC=10						
Reassembled in: 3561173						
ISO/IEC 13818-1 PID=0x64 CC=11						

Figura 3. Captura en wireshark vídeo UDP + aplicación UDP

Toma partido el protocolo no orientado a conexión UDP, es un protocolo no fiable, no controla la gestión ni garantiza el envío de paquetes. UDP no necesita establecer conexión para enviar los paquetes, de ahí que vemos también en la captura protocolos TCP. También podemos ver como se envía el fragmento de vídeo.

2. Emisión de vídeo en UDP + Aplicación TCP

- Indicad las características de la red que muestra la aplicación iPerf.

Para este apartado, he configurado iPerf del siguiente modo:

- Cliente: **iperf -c (IPSERVIDOR)-i 1 -t 600**, donde -c significa que se trata de un cliente, la ip del servidor que en este caso es 212.128.254.47, -i son los segundo en los cuales se volverá a realizar la medición y -t que es el tiempo que va a durar nuestra transición, un tiempo amplio para que dé tiempo a recibir el vídeo.
- Servidor: **iperf -s -i 1**

Ambas máquinas están conectadas al puerto 5001, el tamaño de la ventana TCP es 527 KByte que es lo que se indica por defecto.

- Indicad el comportamiento de la aplicación iPerf antes de la emisión/recepción de vídeo. Vemos el comportamiento de la aplicación antes de iPerf en las siguientes figuras.

```

ylillo@f-l3208-pc07:~$ iperf -c 212.128.254.47 -i 1 -t 600
-----
Client connecting to 212.128.254.47, TCP port 5001
TCP window size: 527 KByte (default)
-----
[ 3] local 212.128.255.71 port 48874 connected with 212.128.254.47 port 5001
[ ID] Interval           Transfer             Bandwidth
[ 3] 0.0 - 1.0 sec      90.1 MBytes        756 Mbits/sec
[ 3] 1.0 - 2.0 sec      88.9 MBytes        746 Mbits/sec
[ 3] 2.0 - 3.0 sec      89.6 MBytes        752 Mbits/sec
[ 3] 3.0 - 4.0 sec      89.5 MBytes        751 Mbits/sec
[ 3] 4.0 - 5.0 sec      89.0 MBytes        747 Mbits/sec
[ 3] 5.0 - 6.0 sec      88.8 MBytes        744 Mbits/sec
[ 3] 6.0 - 7.0 sec      90.5 MBytes        759 Mbits/sec
[ 3] 7.0 - 8.0 sec      88.2 MBytes        740 Mbits/sec
[ 3] 8.0 - 9.0 sec      89.6 MBytes        752 Mbits/sec
[ 3] 9.0 -10.0 sec     88.4 MBytes        741 Mbits/sec

```

Figura 4. Comportamiento cliente en la aplicación iPerf

```

ylillo@f-l3109-pc07:~$ iperf -s -i 1
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 4] local 212.128.254.47 port 5001 connected with 212.128.255.71 port 48874
[ ID] Interval           Transfer             Bandwidth
[ 4] 0.0 - 1.0 sec      88.9 MBytes        746 Mbits/sec
[ 4] 1.0 - 2.0 sec      88.8 MBytes        745 Mbits/sec
[ 4] 2.0 - 3.0 sec      89.4 MBytes        750 Mbits/sec
[ 4] 3.0 - 4.0 sec      89.4 MBytes        750 Mbits/sec
[ 4] 4.0 - 5.0 sec      88.6 MBytes        743 Mbits/sec
[ 4] 5.0 - 6.0 sec      89.2 MBytes        748 Mbits/sec
[ 4] 6.0 - 7.0 sec      89.7 MBytes        752 Mbits/sec
[ 4] 7.0 - 8.0 sec      88.8 MBytes        745 Mbits/sec

```

Figura 5. Comportamiento servidor en la aplicación iPerf

- Indicad el comportamiento de la aplicación iPerf durante la emisión/recepción de vídeo. **Aparentemente vemos que hay mayor calidad de las imágenes, aunque tambien se distorsiona y paraliza muy a menudo.**

- ¿El tráfico de vídeo se ve afectado por el tráfico de iPerf? **Vemos que el ancho de banda se mantiene constante.**

- Adjunta la captura de Wireshark y describe brevemente los paquetes pertenecientes al flujo de la comunicación que se ha desarrollado entre los equipos, mencionando los protocolos que toman partido en la comunicación y los pasos que se llevan a cabo mediante los mensajes de los protocolos.

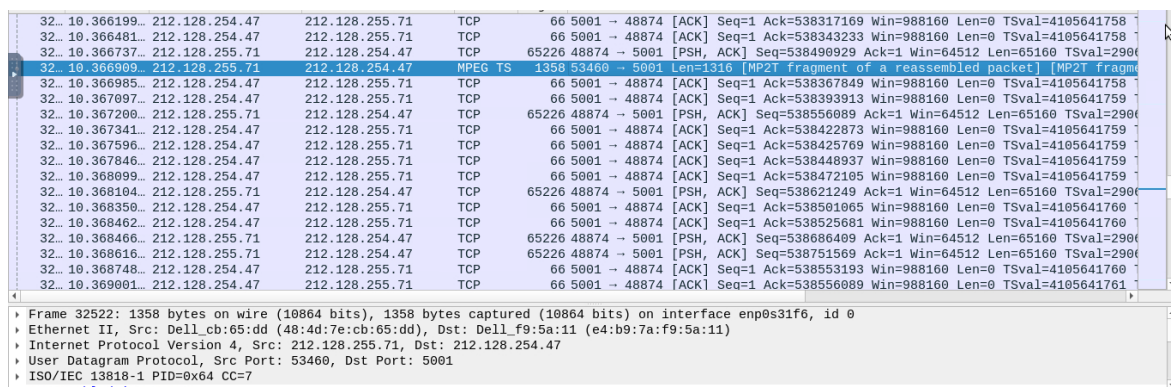


Figura 6. Captura en wireshark vídeo UDP + aplicación TCP

Una vez establecida la conexión, se envía nuestro vídeo, la ventaja es que se transmiten también los paquetes perdidos en caso de que los haya.

3. Emisión de vídeo en RTP + Aplicación UDP

- Indicar las características de la red que muestra la aplicación iPerf.

Las características de la aplicación iPerf son las mismas que en el apartado 1 ya que, también estamos ante una aplicación UDP.

- Indicar el comportamiento de la aplicación iPerf antes de la emisión/recepción de vídeo.

```
ylillo@f-l3208-pc07:~$ iperf -c 212.128.254.47 -u -b 1G -i 1 -t 600
-----
Client connecting to 212.128.254.47, UDP port 5001
Sending 1470 byte datagrams, IPG target: 10.95 us (kalman adjust)
UDP buffer size: 208 KByte (default)
-----
[ 3] local 212.128.255.71 port 39611 connected with 212.128.254.47 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec   113 MBytes  950 Mbits/sec
[ 3] 1.0- 2.0 sec   112 MBytes  938 Mbits/sec
[ 3] 2.0- 3.0 sec   113 MBytes  952 Mbits/sec
[ 3] 3.0- 4.0 sec   113 MBytes  951 Mbits/sec
[ 3] 4.0- 5.0 sec   114 MBytes  952 Mbits/sec
[ 3] 5.0- 6.0 sec   109 MBytes  918 Mbits/sec
```

Figura 7. Comportamiento cliente en la aplicación iPerf

```
ylillo@f-l3109-pc07:~$ iperf -s -u -i 1
-----
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 208 KByte (default)
-----
[ 3] local 212.128.254.47 port 5001 connected with 212.128.255.71 port 39611
[ ID] Interval      Transfer    Bandwidth      Jitter    Lost/Total Datagram
s
[ 3] 0.0- 1.0 sec   113 MBytes  950 Mbits/sec  0.056 ms  0/80798 (0%)
[ 3] 1.0- 2.0 sec   112 MBytes  938 Mbits/sec  0.056 ms  0/78773 (0%)
[ 3] 2.0- 3.0 sec   113 MBytes  952 Mbits/sec  0.035 ms  0/80919 (0%)
[ 3] 3.0- 4.0 sec   113 MBytes  952 Mbits/sec  0.026 ms  0/80929 (0%)
[ 3] 4.0- 5.0 sec   113 MBytes  952 Mbits/sec  0.023 ms  0/80925 (0%)
[ 3] 5.0- 6.0 sec   109 MBytes  918 Mbits/sec  0.027 ms  0/78080 (0%)
```

Figura 8. Comportamiento servidor en la aplicación iPerf

- Indicar el comportamiento de la aplicación iPerf durante la emisión/recepción de vídeo.

Observamos como el cliente de vídeo tarda mucho más en reproducir el contenido, en mi caso, realizo el envío más o menos en el segundo 6 de iPerf. También observamos que aumenta mucho la calidad de la imagen y que no tenemos ni corte ni distorsiones.

- ¿El tráfico de vídeo se ve afectado por el tráfico de iPerf?

Se pierden paquetes pero pocas veces llegan a ser mayores de 0.1%, son inapreciables, el ancho de banda y el jitter se mantienen más o menos constante. Observamos, aunque puede que sea percepción mía, que cuando se para la aplicación iPerf es cuando el cliente reproduce el vídeo.

- ¿Existe alguna diferencia con el vídeo en UDP?

La principal diferencia y más apreciable es la calidad de las imágenes en el video, se ve muchísimo mejor que con UDP, además no tenemos cortes de vídeo ni distorsiones.

- Adjunta la captura de Wireshark y describe brevemente los paquetes pertenecientes al flujo de la comunicación que se ha desarrollado entre los equipos, mencionando los protocolos que toman partido en la comunicación y los pasos que se llevan a cabo mediante los mensajes de los protocolos.

No.	Time	Source	Destination	Protocol	Length	Info
82...	129.64534...	212.128.255.71	212.128.254.47	UDP	1370	59804 → 5001 Len=1328
82...	129.64855...	212.128.255.71	212.128.254.47	UDP	1370	59804 → 5001 Len=1328
82...	129.65175...	212.128.255.71	212.128.254.47	UDP	1370	59804 → 5001 Len=1328
82...	129.65597...	212.128.255.71	212.128.254.47	UDP	1370	59804 → 5001 Len=1328
82...	129.65832...	212.128.255.71	212.128.254.47	UDP	1370	59804 → 5001 Len=1328
82...	129.66303...	212.128.255.71	212.128.254.47	UDP	1370	59804 → 5001 Len=1328
82...	129.66446...	212.128.255.71	212.128.254.47	UDP	1370	59804 → 5001 Len=1328
82...	129.66547...	193.147.79.63	212.128.255.71	TCP	82	58046 → 8000 [PSH, ACK] Seq=171669 Ack=95400505 Win=23569 Len=16 TSval=105938064
82...	129.66599...	212.128.255.71	193.147.79.63	TCP	15061	8000 → 58046 [PSH, ACK] Seq=95400505 Ack=171685 Win=63 Len=14995 TSval=547745
82...	129.66603...	193.147.79.63	212.128.255.71	TCP	82	58046 → 8000 [PSH, ACK] Seq=171685 Ack=95400505 Win=23569 Len=16 TSval=105938064
82...	129.66621...	193.147.79.63	212.128.255.71	TCP	66	58046 → 8000 [ACK] Seq=171701 Ack=95404849 Win=23559 Len=0 TSval=105938064
82...	129.66642...	212.128.255.71	193.147.79.63	TCP	8899	8000 → 58046 [PSH, ACK] Seq=95415500 Ack=171701 Win=63 Len=8833 TSval=547745
82...	129.66647...	193.147.79.63	212.128.255.71	TCP	66	58046 → 8000 [ACK] Seq=171701 Ack=95407745 Win=23559 Len=0 TSval=105938064
82...	129.66673...	193.147.79.63	212.128.255.71	TCP	66	58046 → 8000 [ACK] Seq=171701 Ack=95415500 Win=23552 Len=0 TSval=105938064
82...	129.66673...	193.147.79.63	212.128.255.71	TCP	66	58046 → 8000 [ACK] Seq=171701 Ack=95422740 Win=23559 Len=0 TSval=105938064
82...	129.66770...	212.128.255.71	212.128.254.47	UDP	1370	59804 → 5001 Len=1328
82...	129.67089...	212.128.255.71	212.128.254.47	UDP	1370	59804 → 5001 Len=1328

Frame 8237577: 1370 bytes on wire (10960 bits), 1370 bytes captured (10960 bits) on interface enp0s31f6, id 0
 Ethernet II, Src: Dell_cb:65:dd (48:4d:7e:cb:65:dd), Dst: Dell_f9:5a:11 (e4:b9:7a:f9:5a:11)
 Internet Protocol Version 4, Src: 212.128.255.71, Dst: 212.128.254.47
 User Datagram Protocol, Src Port: 59804, Dst Port: 5001
 Data (1328 bytes)

Figura 9. Captura en wireshark vídeo RTP + aplicación UDP

Recordamos que RTP se utiliza para la transmisión de información en tiempo real y de flujos de media.

UDP es un protocolo sin corrección de errores, y en ningún momento se asegura la llegada de paquetes en su orden de emisión. Para el transporte de datos en tiempo real, como la voz o el vídeo, es necesario utilizar dos protocolos adicionales: RTP (Real-Time Transport Protocol) y RTCP (RTP Control Protocol).

4. Emisión de vídeo en RTP + Aplicación TCP

- Indicad las características de la red que muestra la aplicación iPerf.

Las características de la aplicación iPerf son las mismas que en el apartado 2 ya que, también estamos ante una aplicación TCP.

- Indicad el comportamiento de la aplicación iPerf antes de la emisión/recepción de vídeo.

```

yllillo@f-l3208-pc07:~$ iperf -c 212.128.254.47 -i 1 -t 600
-----
Client connecting to 212.128.254.47, TCP port 5001
TCP window size: 570 KByte (default)
-----
[ 3] local 212.128.255.71 port 48992 connected with 212.128.254.47 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0- 1.0 sec   88.4 MBytes 741 Mbits/sec
[ 3] 1.0- 2.0 sec   87.8 MBytes 736 Mbits/sec
[ 3] 2.0- 3.0 sec   82.1 MBytes 689 Mbits/sec
[ 3] 3.0- 4.0 sec   84.0 MBytes 705 Mbits/sec
[ 3] 4.0- 5.0 sec   86.8 MBytes 728 Mbits/sec
[ 3] 5.0- 6.0 sec   89.4 MBytes 750 Mbits/sec
[ 3] 6.0- 7.0 sec   88.2 MBytes 740 Mbits/sec
[ 3] 7.0- 8.0 sec   89.1 MBytes 748 Mbits/sec

```

Figura 10. Comportamiento cliente en la aplicación iPerf

```

yllillo@f-l3109-pc07:~$ iperf -s -i 1
-----
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
-----
[ 4] local 212.128.254.47 port 5001 connected with 212.128.255.71 port 48992
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0- 1.0 sec   87.2 MBytes 731 Mbits/sec
[ 4] 1.0- 2.0 sec   87.1 MBytes 731 Mbits/sec
[ 4] 2.0- 3.0 sec   82.1 MBytes 689 Mbits/sec
[ 4] 3.0- 4.0 sec   83.9 MBytes 704 Mbits/sec
[ 4] 4.0- 5.0 sec   87.2 MBytes 732 Mbits/sec
[ 4] 5.0- 6.0 sec   89.0 MBytes 746 Mbits/sec
[ 4] 6.0- 7.0 sec   88.3 MBytes 741 Mbits/sec
[ 4] 7.0- 8.0 sec   88.6 MBytes 743 Mbits/sec

```

Figura 11. Comportamiento servidor en la aplicación iPerf

- Indicad el comportamiento de la aplicación iPerf durante la emisión/recepción de vídeo.
Se empieza a reproducir mucho más rápido que en el apartado anterior, llega mucho más rápido, tiene una calidad perfecta además ni se para ni se distorsiona.

- ¿El tráfico de vídeo se ve afectado por el tráfico de iPerf?
El ancho de banda se mantiene constante.

- Adjunta la captura de Wireshark y describe brevemente los paquetes pertenecientes al flujo de la comunicación que se ha desarrollado entre los equipos, mencionando los protocolos que toman partido en la comunicación y los pasos que se llevan a cabo mediante los mensajes de los protocolos.

1460	4.9710355...	193.147.79.63	212.128.255.71	TCP	82 58046 → 8000 [PSH, ACK] Seq=6097 Ack=2478958 Win=23569 Len=16 TSval=105846
1461	4.9712426...	212.128.255.71	193.147.79.63	TCP	66 8000 → 58046 [ACK] Seq=2478958 Ack=6113 Win=0 TSval=546288692 TSecr=
1462	4.9994019...	ExtremeNetworks.d7...	LLDP Multicast	LLDP	60 80/02:04:96:d7:82:a7 1W/142 120
1463	5.0003234...	JuniperN_f1:e7:f0	Broadcast	ARP	60 Who has 212.128.254.181? Tell 212.128.254.1
1464	5.0009771...	193.147.79.63	212.128.255.71	TCP	84 58046 → 8000 [PSH, ACK] Seq=6113 Ack=2478958 Win=23569 Len=18 TSval=105846
1465	5.0029258...	212.128.255.71	193.147.79.63	TCP	120 8000 → 58046 [PSH, ACK] Seq=2478958 Ack=6131 Win=63 Len=54 TSval=546288724
1466	5.0078807...	212.128.255.71	212.128.254.47	TCP	74 48992 → 5001 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=290814487
1467	5.0085123...	212.128.254.47	212.128.255.71	TCP	74 5001 → 48992 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1
1468	5.0085866...	212.128.255.71	212.128.254.47	TCP	66 48992 → 5001 [ACK] Seq=1 Ack=1 Win=64512 Len=0 TSval=2908144874 TSecr=4107507304
1469	5.0090229...	212.128.255.71	212.128.254.47	TCP	7306 48992 → 5001 [PSH, ACK] Seq=1 Ack=1 Win=64512 Len=7240 TSval=2908144875 TSecr=
1470	5.0090475...	212.128.255.71	212.128.254.47	TCP	7306 48992 → 5001 [PSH, ACK] Seq=7241 Ack=1 Win=64512 Len=7240 TSval=2908144875 TSecr=
1471	5.0093841...	212.128.254.47	212.128.255.71	TCP	66 5001 → 48992 [ACK] Seq=1 Ack=7241 Win=61440 Len=0 TSval=4107507304 TSecr=
1472	5.0093843...	212.128.254.47	212.128.255.71	TCP	66 5001 → 48992 [ACK] Seq=1 Ack=8689 Win=60416 Len=0 TSval=4107507304 TSecr=
1473	5.0094228...	212.128.255.71	212.128.254.47	TCP	10202 48992 → 5001 [PSH, ACK] Seq=14481 Ack=1 Win=64512 Len=10136 TSval=2908144875 TSecr=
1474	5.0094478...	212.128.254.47	212.128.255.71	TCP	66 5001 → 48992 [ACK] Seq=1 Ack=14481 Win=56320 Len=0 TSval=4107507304 TSecr=
1475	5.0094596...	212.128.255.71	212.128.254.47	TCP	14546 48992 → 5001 [PSH, ACK] Seq=24617 Ack=1 Win=64512 Len=14480 TSval=2908144875 TSecr=
1476	5.0094695...	212.128.255.71	212.128.254.47	TCP	4410 48992 → 5001 [PSH, ACK] Seq=39097 Ack=1 Win=64512 Len=4344 TSval=2908144875 TSecr=
1477	5.0096947...	212.128.254.47	212.128.255.71	TCP	66 5001 → 48992 [ACK] Seq=1 Ack=21721 Win=51200 Len=0 TSval=4107507305 TSecr=

Frame 1470: 7306 bytes on wire (58448 bits), 7306 bytes captured (58448 bits) on interface enp0s31f6, id 0
 Ethernet II, Src: Dell_cb:65:dd (48:4d:7e:cb:65:dd), Dst: Dell_f9:5a:11 (e4:b9:7a:f9:5a:11)
 Internet Protocol Version 4, Src: 212.128.255.71, Dst: 212.128.254.47
 Transmission Control Protocol, Src Port: 48992, Dst Port: 5001, Seq: 7241, Ack: 1, Len: 7240
 Data (7240 bytes)

Figura 12. Captura en wireshark vídeo RTP + aplicación TCP

TCP retransmite los paquetes perdidos, potencialmente tenemos la capacidad de proporcionar una calidad de sonido superior que con UDP. De modo general, en la captura general podemos ver el establecimiento de la conexión(con intercambio de números de secuencia), la transferencia de datos, recordamos que tenemos un checksum constante y finalizamos la conexión.