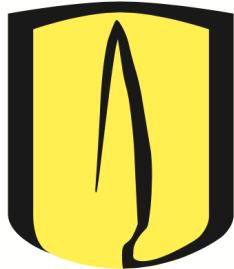


Universidad de los Andes

Departamento de Ingeniería de Sistemas



**Laboratorio #3: Análisis De Capa de Transporte y
Sockets**

ISIS3204 - Infraestructura de Comunicaciones

Grupo 3:

Juan Esteban Quiroga - 202013216

Juan Manuel Rodriguez - 202013372

Andres Felipe Ortiz - 201727662

2025-10

Contents

Introducción	3
1 Modelo Publicador–Suscriptor	4
1.1 Configuración	4
1.1.1 Compilación	4
1.1.2 Ejecución	5
1.2 Publicador	5
1.2.1 Descripción	5
1.2.2 Tipos de eventos	6
1.2.3 Reglas de negocio	6
1.3 Broker	6
1.3.1 Descripción	6
1.4 Suscriptor	6
1.4.1 Descripción	6
2 Pruebas y comparación	7
2.1 TCP	7
2.1.1 Conexión suscriptores a broker	7
2.1.2 Transmisión de eventos	9
2.2 UDP	13
3 Preguntas de análisis	18
4 Tabla comparativa	18
5 Análisis de Resultados del Laboratorio	18
5.1 ¿Qué ocurriría si en lugar de dos publicadores hubiera cien partidos simultáneos?	18
5.2 Si un gol se envía como mensaje y un suscriptor no lo recibe en UDP, ¿qué implicaciones tendría?	18
5.3 Protocolo más adecuado para seguimiento en vivo de partidos	18
5.4 Comparación de overhead entre TCP y UDP	18
5.5 Efecto de desorden en UDP y posibles soluciones	19
5.6 Efecto del número de suscriptores sobre el desempeño	19
5.7 Comportamiento ante caída del broker	19
5.8 Sincronización simultánea de actualizaciones críticas	19
5.9 Uso de CPU y memoria en el broker	19
5.10 Diseño de un sistema real para millones de usuarios	19

Introducción

En este laboratorio implementamos un sistema de mensajería usando el modelo publicador-suscriptor con sockets TCP y UDP para simular la transmisión en tiempo real de eventos en partidos de la UEFA Champions League. Los publicadores actúan como periodistas que envían actualizaciones en vivo de los partidos, el broker gestiona la distribución de los mensajes por partido, y los suscriptores reciben las notificaciones relevantes a las que están suscritos. El sistema demuestra la comunicación asíncrona, la escalabilidad y los compromisos entre confiabilidad y rendimiento de los protocolos de transporte TCP y UDP en aplicaciones basadas en sockets.

Escenario de prueba

Nosotros creamos un sistema que permite generar simulaciones de partidos entre los siguientes equipos:

- Real Madrid CF
- Juventus FC
- Paris Saint-Germain FC
- FC Barcelona
- Liverpool FC
- FC Bayern Munich

Sin embargo, para las pruebas realizadas en Wireshark, montamos 3 partidos en particular: `RealMadrid_vs_Juventus`, `Liverpool_vs_BayernMunich`, y `Barcelona_vs_PSG` en los cuales hay un publicador por partido transmitiendo al broker. Por otro lado tenemos 4 suscriptores:

- Suscriptor #1: `RealMadrid_vs_Juventus`, `Liverpool_vs_BayernMunich`, y `Barcelona_vs_PSG`
- Suscriptor #2: `RealMadrid_vs_Juventus`
- Suscriptor #3: `Liverpool_vs_BayernMunich`
- Suscriptor #4: `Barcelona_vs_PSG` y `Liverpool_vs_BayernMunich`

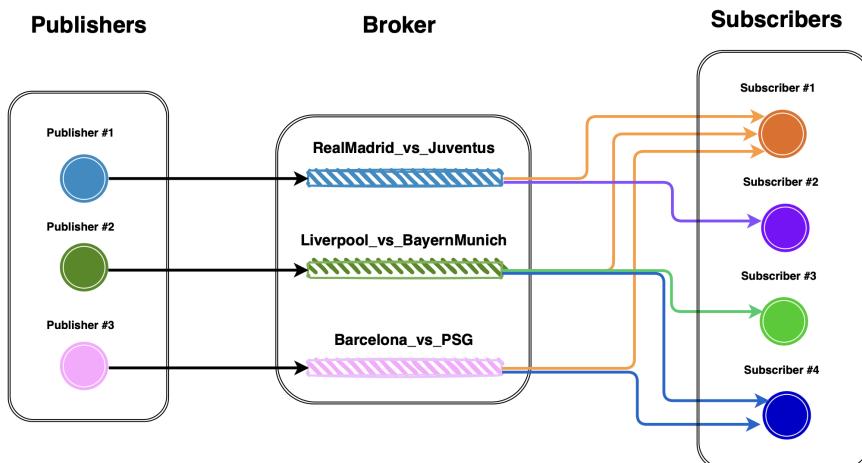


Figure 1: Configuración de escenario de prueba

1 Modelo Publicador–Suscriptor

1.1 Configuración

Nuestro proyecto tiene la siguiente configuración que fue diseñada para poder reutilizar código compartido entre los archivos TCP y UDP.

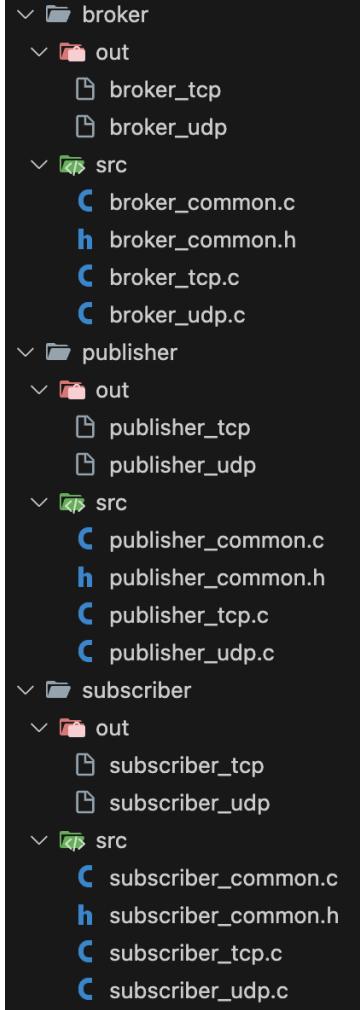


Figure 2: Estructura de proyecto

1.1.1 Compilación

```
gcc broker/src/broker_common.c broker/src/broker_tcp.c -o broker/out/broker_tcp -Wall -Wextra -std=c11 -O2

gcc broker/src/broker_common.c broker/src/broker_udp.c -o broker/out/broker_udp -Wall -Wextra -std=c11 -O2

gcc subscriber/src/subscriber_common.c subscriber/src/subscriber_tcp.c -o subscriber/out/subscriber_tcp -Wall -Wextra -std=c11 -O2

gcc subscriber/src/subscriber_common.c subscriber/src/subscriber_udp.c -o subscriber/out/subscriber_udp -Wall -Wextra -std=c11 -O2

gcc publisher/src/publisher_common.c publisher/src/publisher_tcp.c -o publisher/out/publisher_tcp -Wall -Wextra -std=c11 -O2

gcc publisher/src/publisher_common.c publisher/src/publisher_udp.c -o publisher/out/publisher_udp -Wall -Wextra -std=c11 -O2
```

1.1.2 Ejecución

Cada uno se corre en su propio tab en la terminal...

Broker

```
./broker/out/broker_tcp 5050
```

Subscriber 1

```
./subscriber/out/subscriber_tcp 127.0.0.1 5050 RealMadrid_vs_Juventus Liverpool_vs_BayernMunich  
Barcelona_vs_PSG
```

Subscriber 2

```
./subscriber/out/subscriber_tcp 127.0.0.1 5050 RealMadrid_vs_Juventus
```

Subscriber 3

```
./subscriber/out/subscriber_tcp 127.0.0.1 5050 Liverpool_vs_BayernMunich
```

Subscriber 4

```
./subscriber/out/subscriber_tcp 127.0.0.1 5050 Barcelona_vs_PSG Liverpool_vs_BayernMunich
```

Publisher 1

```
./publisher/out/publisher_tcp 127.0.0.1 5050 RealMadrid_vs_Juventus --demo
```

Publisher 2

```
./publisher/out/publisher_tcp 127.0.0.1 5050 Liverpool_vs_BayernMunich --demo
```

Publisher 3

```
./publisher/out/publisher_udp 127.0.0.1 5050 Barcelona_vs_PSG --demo
```

1.2 Publicador

1.2.1 Descripción

El publicador es el componente encargado de generar y enviar los mensajes correspondientes a los eventos de cada partido. Existen dos versiones: una basada en TCP y otra en UDP, que utilizan el mismo módulo de simulación pero son distintos en la forma en que transmiten los datos.

- En la versión TCP, el publicador establece una conexión confiable con el broker mediante un socket orientado a conexión. Una vez conectado, envía los eventos de forma secuencial usando la función `sendTo()`, garantizando que los mensajes lleguen en orden y sin pérdidas.
- En la versión UDP, el publicador utiliza un socket sin conexión, enviando cada evento mediante la función `send()` al puerto del broker especificado. Es necesario usar esta función ya que UDP no establece un 3-way handshake entonces hay que identificar el suscriptor para enviar el evento). Aunque este método no asegura la entrega o el orden de los mensajes, permite una transmisión más rápida y ligera, simulando escenarios donde la inmediatez es prioritaria sobre la confiabilidad.

En ambos casos, los mensajes se formatean con el tema del partido y el tipo de evento, y se envían con un intervalo de aproximadamente un segundo entre ellos.

Cada mensaje del publicador sigue el siguiente formato:

```
<Match_name>| [EVENT_TYPE] <Event description> at <minute>' -- <TeamA> <scoreA>-<scoreB> <TeamB>
```

1.2.2 Tipos de eventos

Estructura de partido:

Cada partido dura 90 minutos y es dividido en 12 eventos. Para cada evento del partido, se elige un tiempo aleatorio entre rangos de valores de cada "bucket" de tiempo. Por ejemplo, el primer bucket incluye minutos del [1-7] entonces estos son posibles valores para lanzar el primer evento del partido. Cada uno de los 10 eventos del partido puede pertenecer a una de las siguientes categorías:

- **[GOAL]** – algún jugador anota un gol y otro jugador marca la asistencia
- **[CARD]** – algún jugador recibe tarjeta amarilla o tarjeta roja
- **[SUB]** – algún jugador que está jugando en el campo es reemplazado por algún jugador que está en la banca del mismo equipo y aún no ha jugado en ese partido. (solo sucede después de 55')
- **[HT]** – indica el marcador en el mediotiempo (45')
- **[FT]** – indica el marcador final (90')

1.2.3 Reglas de negocio

Tarjetas

- La primera vez que un jugador es señalado, se le muestra la tarjeta amarilla
- La segunda vez que un jugador es señalado, se le muestra la tarjeta roja y es expulsado del partido dejando al equipo con 10 hombres en el campo
- 20% de las veces en cualquier evento, a un jugador se le puede mostrar una tarjeta roja directa

Anotar

- El sistema elige uno de los dos equipos para anotar cuando se genera un evento de tipo [GOAL]
- El jugador solo se elige si no ha sido expulsado (igual que el jugador que da la asistencia)

1.3 Broker

1.3.1 Descripción

El broker actúa como un intermediario entre los publicadores y los suscriptores dentro del sistema de notificaciones de eventos. El broker recibe los mensajes enviados por los publicadores (a través de TCP o UDP), identifica el partido al que pertenecen y los reenvía únicamente a los suscriptores que estén suscritos a ese partido.

Cada suscriptor envía un mensaje al broker que indica el interés en un partido. El broker guarda esa suscripción en una tabla interna de partidos y cada vez que llega un mensaje con ese mismo nombre de partido, lo reenvía a todos los suscriptores asociados.

(El broker no contiene lógica de negocio, ya que su función se limita al enrutamiento de mensajes. No interpreta el contenido de los eventos, ya que solo actúa como un intermediario que se encarga de que cada suscriptor reciba únicamente la información correspondiente a los partidos a los que está suscrito.)

1.4 Suscriptor

1.4.1 Descripción

El suscriptor representa al cliente final que recibe los eventos transmitidos por el sistema. Su función principal es conectarse al broker (mediante TCP o UDP) y registrarse en uno o varios temas específicos, enviando un mensaje con el siguiente formato:

```
SUBSCRIBE|<topic_1> <topic_2> ... <topic_n>
```

Una vez suscrito, el suscriptor permanece a la espera de mensajes provenientes del broker. Cada mensaje recibido tiene información sobre un evento en tiempo real que corresponde a los partidos a los que se ha suscrito.

En la versión TCP, el suscriptor mantiene una conexión persistente con el broker y recibe los mensajes mediante `recv()`. En la versión UDP, no existe una conexión formal, sino que los mensajes llegan directamente al puerto local asociado al socket mediante `recvfrom()`.

2 Pruebas y comparación

En esta sección, verificamos el funcionamiento del modelo de comunicación.

2.1 TCP

TCP es el protocolo que tiene envío de paquetes confiable, los mensajes llegan ordenados, y requiere una conexión persistente que se establece con un handshake. El flujo que sigue nuestra aplicación se describe a continuación:

1. El broker crea un socket y escucha en un puerto esperando conexiones de publicadores y suscriptores.
2. Los suscriptores se conectan al broker por medio de conexión TCP usando el 3-way handshake (SYN, SYN-ACK, ACK).
3. El suscriptor manda un mensaje al broker que contiene las suscripciones que quiere.
4. Los publicadores establecen una conexión TCP con el broker.
5. Los publicadores envían eventos de partidos al broker.
6. El broker identifica los suscriptores que están suscritos a los partidos asociados a los eventos que llegan de los publicadores.
7. El broker envía los mensajes de los eventos por medio de la conexión TCP a los suscriptores.
8. El suscriptor recibe y muestra el mensaje de cada evento.
9. Se cierran las conexiones de los publicadores automáticamente después de enviar el último evento (los suscriptores siguen escuchando hasta desconectarse manualmente).

2.1.1 Conexión suscriptores a broker

Primero, vamos a demostrar que los suscriptores establecen un 3-way handshake con el broker. En los screenshots vemos que el suscriptor #1 y el suscriptor #3 hace bind con un puerto aleatorio, establece el handshake, y envía mensajes al broker indicando los partidos a los que se quiere suscribir. (para evitar redundancia solo se analizan los suscriptores 1 y 3)

En las captura de Wireshark, los segmentos TCP con las banderas PSH y ACK indican que el emisor está enviando datos que deben entregarse inmediatamente a la aplicación receptora, sin esperar más bytes en el búfer. El campo Seq (secuencia) muestra el número del primer byte transmitido en ese segmento, mientras que Ack (acuse) confirma los bytes recibidos del otro extremo. Por ejemplo, un paquete con Seq=1 seguido de otro con Seq=34 significa que el primer mensaje contenía 33 bytes y que el segundo continúa justo después, reflejando la naturaleza secuencial y confiable de la transmisión TCP.

Suscriptor #1 (Puerto 64376)

tcp.port == 64376					
Time	Source	Destination	Protocol	Length	Info
1 0.000000	127.0.0.1	127.0.0.1	TCP	68	64376 -> 5050 [SYN] Seq=0 Win=65535 Len=0 MSS=64 TSval=1702896057 TSecr=0 SACK_PERM
2 0.000105	127.0.0.1	127.0.0.1	TCP	68	5050 -> 64376 [SYN, ACK] Seq=1 Win=65535 Len=0 MSS=64 TSval=3299373255 TSecr=1702896057 SACK_PERM
3 0.000126	127.0.0.1	127.0.0.1	TCP	56	64376 -> 5050 [ACK] Seq=1 Ack=1 Win=408320 Len=0 TSval=1702896057 TSecr=3299373255
4 0.000143	127.0.0.1	127.0.0.1	TCP	56	[TCP Window Update] 5050 -> 64376 [ACK] Seq=1 Ack=1 Win=408320 Len=0 TSval=3299373255 TSecr=1702896057

Figure 3: Conexión de suscriptor 1

```
> ./subscriber/out/subscriber_tcp 127.0.0.1 5050 RealMadrid_vs_Juventus Liverpool_vs_BayernMunich Barcelona_vs_PSG
[Subscriber-TCP] Connected to broker 127.0.0.1:5050
[Subscriber-TCP] Subscribed to topic: RealMadrid_vs_Juventus | Local port: 64376
[Subscriber-TCP] Subscribed to topic: Liverpool_vs_BayernMunich | Local port: 64376
[Subscriber-TCP] Subscribed to topic: Barcelona_vs_PSG | Local port: 64376
```

Figure 4: Verificación de conexión de suscriptor 1 en la terminal de suscriptor

[64376 → 5050 [PSH, ACK] Seq=1

02 00 00 00 45 00 00 55 00 00 40 00 40 06 00 00E..U ..@@..
7f 00 00 01 7f 00 00 01 fb 78 13 ba 61 5f 3e 8dx..a_>..
0c 9a fc bd 80 18 18 ec fe 49 00 00 01 01 08 0aI.....
65 80 21 b9 c4 a8 70 c7 53 55 42 53 43 52 49 42	e..!...p.. SUBSCRIB
45 7c 52 65 61 6c 4d 61 64 72 69 64 5f 76 73 5f	E RealMa drid_vs_
4a 75 76 65 6e 74 75 73 0a	Juventus ..

[64376 → 5050 [PSH, ACK] Seq=34

02 00 00 00 45 00 00 73 00 00 40 00 40 06 00 00E..s ..@@..
7f 00 00 01 7f 00 00 01 fb 78 13 ba 61 5f 3e aex..a_>..
0c 9a fc bd 80 18 18 ec fe 67 00 00 01 01 08 0ag.....
65 80 21 b9 c4 a8 70 c7 53 55 42 53 43 52 49 42	e..!...p.. SUBSCRIB
45 7c 4c 69 76 65 72 70 6f 6f 6c 5f 76 73 5f 42	E Liverp ool_vs_B
61 79 65 72 6e 4d 75 6e 69 63 68 0a 53 55 42 53	ayernMun ich-SUBS
43 52 49 42 45 7c 42 61 72 63 65 6c 6f 6e 61 5f	CRIBE Ba rcelona_
76 73 5f 50 53 47 0a	vs_PSG..

Figure 5: Mensaje de suscripción al broker

```
> ./broker/out/broker_udp 5050
[Broker-UDP] Listening on port 5050...
[Broker] Created new topic #0: RealMadrid_vs_Juventus
[Broker-UDP] Subscriber 127.0.0.1:49622 subscribed to 'RealMadrid_vs_Juventus' (Topic #0, 1 total subs)
[Broker] Created new topic #1: Liverpool_vs_BayernMunich
[Broker-UDP] Subscriber 127.0.0.1:49622 subscribed to 'Liverpool_vs_BayernMunich' (Topic #1, 1 total subs)
[Broker] Created new topic #2: Barcelona_vs_PSG
```

Figure 6: Suscripciones de suscriptor 1 en terminal del broker

Suscriptor #3 (Puerto 64378)

tcp.port == 64378						
No.	Time	Source	Destination	Protocol	Length	Info
15	3.825482	127.0.0.1	127.0.0.1	TCP	68	64378 → 5050 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=3535411727 TSecr=0 SACK_PERM
16	3.825683	127.0.0.1	127.0.0.1	TCP	68	5050 → 64378 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=2462472733 TSecr=3535411727 SACK_PERM
17	3.825744	127.0.0.1	127.0.0.1	TCP	56	64378 → 5050 [ACK] Seq=1 Ack=1 Win=408320 Len=0 TSval=3535411727 TSecr=2462472733
18	3.825772	127.0.0.1	127.0.0.1	TCP	92	64378 → 5050 [PSH, ACK] Seq=1 Ack=1 Win=408320 Len=36 TSval=3535411727 TSecr=2462472733

Figure 7: Conexión de suscriptor 3

```
> ./subscriber/out/subscriber_tcp 127.0.0.1 5050 Liverpool_vs_BayernMunich
[Subscriber-TCP] Connected to broker 127.0.0.1:5050
[Subscriber-TCP] Subscribed to topic: Liverpool_vs_BayernMunich | Local port: 64378
```

Figure 8: Verificación de conexión de suscriptor 3 en la terminal de suscriptor

[64378 → 5050 [PSH, ACK] Seq=1 Ack=1

02 00 00 00 45 00 00 58	00 00 40 00 40 06 00 00E..X..@..@..
7f 00 00 01 7f 00 00 01	fb 7a 13 ba 84 66 dd c7z...f..
e8 67 c9 ac 80 18 18 ec	fe 4c 00 00 01 01 08 0a	g.....L.....
d2 ba 1a 0f 92 c6 5a 1d	53 55 42 53 43 52 49 42Z..SUBSCRIB
45 7c 4c 69 76 65 72 70	6f 6f 6c 5f 76 73 5f 42	E Liverp ool_vs_B
61 79 65 72 6e 4d 75 6e	69 63 68 0a	ayernMun ich..

Figure 9: Mensaje de suscripción al broker

```
[Broker-TCP] Client 7 subscribed to 'Liverpool_vs_BayernMunich' (Topic #1, 2 total subs, Client now follows 1 topics)
[Broker-TCP] New connection from 127.0.0.1:64379 (fd=8)
```

Figure 10: Suscripciones de suscriptor 3 en terminal del broker

Cada vez que el broker detecta un partido nuevo, crea una fila en la tabla interna y si llega un suscriptor futuro que quiere recibir eventos de ese partido, solo lo asigna en la tabla.

Luego de establecer la conexión y registrar los partidos que quiere escuchar, el broker queda pendiente de mensajes con eventos de los publicadores.

2.1.2 Transmisión de eventos

Como se menciona anteriormente, cada publicador también establece una conexión TCP por medio de 3-way handshake y luego envía los eventos al broker. (Por simplicidad, solo voy a detallar en Wireshark los mensajes enviados de un publicador pero al final muestro los eventos que cada publicador envió)

Publicador #1 (Puerto 64380)

tcp.port == 64380						
No.	Time	Source	Destination	Protocol	Length	Info
29	11.771882	127.0.0.1	127.0.0.1	TCP	68	64380 → 5050 [SYN] Seq=0 Win=65535 Len=0 MSS=16344 WS=64 TSval=1620361170 TSecr=0 SACK_PERM
30	11.771983	127.0.0.1	127.0.0.1	TCP	68	5050 → 64380 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=16344 WS=64 TSval=3257512708 TSecr=1620361170 SACK_PERM
31	11.771923	127.0.0.1	127.0.0.1	TCP	56	64380 → 5050 [ACK] Seq=1 Ack=1 Win=408320 Len=0 TSval=1620361170 TSecr=3257512708

Figure 11: Conexión de publicador 1 al broker

Para evitar redundancia en el informe, adjunto un sampleo de 3 eventos (de los 12) para el partido Real Madrid vs Juventus.

02 00 00 00 00 45 00 00 96	00 00 40 00 40 06 00 00E.....@@..
7f 00 00 01 7f 00 00 01	fb 7c 13 ba 1e ab a2 6aj
44 40 0a 3d 80 18 18 ec	fe 8a 00 00 01 01 08 0a	D@=.....
60 94 bf d2 c2 29 b3 04	52 65 61 6c 4d 61 64 72	`....)..\ RealMadr
69 64 5f 76 73 5f 4a 75	76 65 6e 74 75 73 7c 5b	id_vs_Ju ventus [
47 4f 41 4c 5d 20 52 61	62 69 6f 74 20 73 63 6f	GOAL] Ra biot sco
72 65 73 20 28 61 73 73	69 73 74 3a 20 4c 6f 63	res (ass ist: Loc
61 74 65 6c 6c 69 29 20	61 74 20 36 27 20 e2 80	atelli) at 6' ..
94 20 52 65 61 6c 4d 61	64 72 69 64 20 30 2d 31	· RealMa drid 0-1
20 4a 75 76 65 6e 74 75	73 0a	Juventu s·

02 00 00 00 45 00 00 9a	00 00 40 00 40 06 00 00E.....@@..
7f 00 00 01 7f 00 00 01	fb 7c 13 ba 1e ab a5 59Y
44 40 0a 3d 80 18 18 ec	fe 8e 00 00 01 01 08 0a	D@=.....
60 94 e3 17 c2 29 d2 5c	52 65 61 6c 4d 61 64 72	`....)..\ RealMadr
69 64 5f 76 73 5f 4a 75	76 65 6e 74 75 73 7c 5b	id_vs_Ju ventus [
47 4f 41 4c 5d 20 43 6f	75 72 74 6f 69 73 20 73	GOAL] Co urtois s
63 6f 72 65 73 20 28 61	73 73 69 73 74 3a 20 42	cores (a ssist: B
65 6c 6c 69 6e 67 68 61	6d 29 20 61 74 20 36 35	ellingha m) at 65
27 20 e2 80 94 20 52 65	61 6c 4d 61 64 72 69 64	' ... Re alMadrid
20 31 2d 34 20 4a 75 76	65 6e 74 75 73 0a	1-4 Juv entus·

02 00 00 00 45 00 00 7a	00 00 40 00 40 06 00 00E.....z ..@@..
7f 00 00 01 7f 00 00 01	fb 7c 13 ba 1e ab a5 ff
44 40 0a 3d 80 18 18 ec	fe 6e 00 00 01 01 08 0a	D@=.....n.....
60 94 ea ed c2 29 da 33	52 65 61 6c 4d 61 64 72	`....)..\3 RealMadr
69 64 5f 76 73 5f 4a 75	76 65 6e 74 75 73 7c 5b	id_vs_Ju ventus [
46 54 5d 20 39 30 27 20	46 75 6c 6c 20 74 69 6d	FT] 90' Full tim
65 20 e2 80 94 20 52 65	61 6c 4d 61 64 72 69 64	e ... Re alMadrid
20 31 2d 34 20 4a 75 76	65 6e 74 75 73 0a	1-4 Juv entus·

Figure 12: Mensajes enviados al broker

En la siguiente gráfica, muestro el log del broker de los mensajes que van llegando de los publicadores y se reenvían a los suscriptores..

```
[Broker-TCP] New connection from 127.0.0.1:64380 (fd=9)
[Broker-TCP] Forwarded to 2 subscriber(s) for RealMadrid_vs_Juventus
[Broker-TCP] New connection from 127.0.0.1:64381 (fd=10)
[Broker-TCP] Forwarded to 3 subscriber(s) for Liverpool_vs_BayernMunich
[Broker-TCP] New connection from 127.0.0.1:64382 (fd=11)
[Broker-TCP] Forwarded to 2 subscriber(s) for Barcelona_vs_PSG
[Broker-TCP] Forwarded to 2 subscriber(s) for RealMadrid_vs_Juventus
[Broker-TCP] Forwarded to 3 subscriber(s) for Liverpool_vs_BayernMunich
[Broker-TCP] Forwarded to 2 subscriber(s) for Barcelona_vs_PSG
[Broker-TCP] Forwarded to 2 subscriber(s) for RealMadrid_vs_Juventus
[Broker-TCP] Forwarded to 3 subscriber(s) for Liverpool_vs_BayernMunich
[Broker-TCP] Forwarded to 2 subscriber(s) for Barcelona_vs_PSG
[Broker-TCP] Forwarded to 2 subscriber(s) for RealMadrid_vs_Juventus
[Broker-TCP] Forwarded to 3 subscriber(s) for Liverpool_vs_BayernMunich
[Broker-TCP] Forwarded to 2 subscriber(s) for Barcelona_vs_PSG
[Broker-TCP] Forwarded to 2 subscriber(s) for RealMadrid_vs_Juventus
[Broker-TCP] Forwarded to 3 subscriber(s) for Liverpool_vs_BayernMunich
[Broker-TCP] Forwarded to 2 subscriber(s) for Barcelona_vs_PSG
```

Figure 13: Broker logs

En lo próxima captura, se ven todos los paquetes que recibe suscriptor 1 de los varios publicadores (mensajes despachados por el broker)

237	21.678658	127.0.0.1	127.0.0.1	TCP	144	5850 - 64376 [PSH, ACK] Seq=2486 Ack=97 Win=408320 Len=88 TStamp=3209384934 TSect=1702017227
239	21.678782	127.0.0.1	127.0.0.1	TCP	56	64376 - 5859 [ACK] Seq=97 Ack=2574 Win=408720 Len=88 TStamp=1702017738 TSect=3209394934
243	21.683280	127.0.0.1	127.0.0.1	TCP	110	5850 - 64376 [PSH, ACK] Seq=2574 Ack=98 Win=408320 Len=88 TStamp=1702017738 TSect=3209394934
246	21.683618	127.0.0.1	127.0.0.1	TCP	56	64376 - 5858 [ACK] Seq=97 Ack=2637 Win=408720 Len=88 TStamp=1702017766 TSect=3209394934
249	22.173658	127.0.0.1	127.0.0.1	TCP	125	5850 - 64376 [PSH, ACK] Seq=2637 Ack=97 Win=408320 Len=88 TStamp=1702017766 TSect=3209394934
252	22.173736	127.0.0.1	127.0.0.1	TCP	56	64376 - 5858 [ACK] Seq=97 Ack=2786 Win=408656 Len=88 TStamp=1702018231 TSect=3209394934
257	22.682121	127.0.0.1	127.0.0.1	TCP	115	5850 - 64376 [PSH, ACK] Seq=2786 Ack=97 Win=408320 Len=88 TStamp=1702018231 TSect=3209394934
259	22.682209	127.0.0.1	127.0.0.1	TCP	56	64376 - 5858 [ACK] Seq=97 Ack=2785 Win=408656 Len=88 TStamp=1702018739 TSect=3209394934
263	22.886698	127.0.0.1	127.0.0.1	TCP	125	5850 - 64376 [PSH, ACK] Seq=2785 Ack=97 Win=408320 Len=88 TStamp=1702018739 TSect=3209394934
268	22.897600	127.0.0.1	127.0.0.1	TCP	56	64376 - 5858 [ACK] Seq=97 Ack=2834 Win=408320 Len=88 TStamp=1702018864 TSect=3209394934
269	23.173447	127.0.0.1	127.0.0.1	TCP	131	5850 - 64376 [PSH, ACK] Seq=2834 Ack=97 Win=408320 Len=88 TStamp=1702018864 TSect=3209394934
272	23.177215	127.0.0.1	127.0.0.1	TCP	56	64376 - 5858 [ACK] Seq=97 Ack=2989 Win=408528 Len=57 TStamp=1702019234 TSect=3209394934
277	23.687066	127.0.0.1	127.0.0.1	TCP	113	5850 - 64376 [PSH, ACK] Seq=2989 Ack=97 Win=408528 Len=57 TStamp=1702019234 TSect=3209394934
279	23.687130	127.0.0.1	127.0.0.1	TCP	56	64376 - 5858 [ACK] Seq=97 Ack=2986 Win=406528 Len=88 TStamp=1702019744 TSect=3209394934

Figure 14: Mensajes recibidos por suscriptor 1

En las capturas siguientes, se ven los mensajes generados por los publicadores:

```
./publisher/out/publisher_tcp 127.0.0.1 5050 RealMadrid_vs_Juventus --demo
[TCP] Connected to 127.0.0.1:5050
Sent: RealMadrid_vs_Juventus|[GOAL] Rabiot scores (assist: Locatelli) at 6' - RealMadrid 0-1 Juventus
Sent: RealMadrid_vs_Juventus|[RED] Direct red card for Rudiger at 12 (Player expelled)
Sent: RealMadrid_vs_Juventus|[GOAL] Vlahovic scores (assist: Szczesny) at 18' - RealMadrid 0-2 Juventus
Sent: RealMadrid_vs_Juventus|[GOAL] McKennie scores (assist: Locatelli) at 23' - RealMadrid 0-3 Juventus
Sent: RealMadrid_vs_Juventus|[GOAL] Rabiot scores (assist: Bremer) at 35' - RealMadrid 0-4 Juventus
Sent: RealMadrid_vs_Juventus|[CARD] Yellow card for Valverde at 38'
Sent: RealMadrid_vs_Juventus|[HT] 45' Halftime - RealMadrid 0-4 Juventus
Sent: RealMadrid_vs_Juventus|[CARD] Yellow card for McKennie at 54'
Sent: RealMadrid_vs_Juventus|[RED] Direct red card for Valverde at 61 (Player expelled)
Sent: RealMadrid_vs_Juventus|[GOAL] Courtois scores (assist: Bellingham) at 65' - RealMadrid 1-4 Juventus
Sent: RealMadrid_vs_Juventus|[SUB] Vinicius replaced by Modric at 77'
Sent: RealMadrid_vs_Juventus|[FT] 90' Full time - RealMadrid 1-4 Juventus
[TCP] Connection closed.
```

Figure 15: Real Madrid vs Juventus

```
./publisher/out/publisher_tcp 127.0.0.1 5050 Liverpool_vs_BayernMunich --demo
[TCP] Connected to 127.0.0.1:5050
Sent: Liverpool_vs_BayernMunich|[GOAL] Kimmich scores (assist: Sane) at 6' - Liverpool 0-1 BayernMunich
Sent: Liverpool_vs_BayernMunich|[GOAL] Szoboszlai scores (assist: Mac Allister) at 12' - Liverpool 1-1 BayernMunich
Sent: Liverpool_vs_BayernMunich|[GOAL] Alisson scores (assist: Salah) at 19' - Liverpool 2-1 BayernMunich
Sent: Liverpool_vs_BayernMunich|[GOAL] TAA scores (assist: Alisson) at 22' - Liverpool 3-1 BayernMunich
Sent: Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Mac Allister) at 34' - Liverpool 4-1 BayernMunich
Sent: Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Mac Allister) at 37' - Liverpool 5-1 BayernMunich
Sent: Liverpool_vs_BayernMunich|[HT] 45' Halftime - Liverpool 5-1 BayernMunich
Sent: Liverpool_vs_BayernMunich|[CARD] Yellow card for Kimmich at 55'
Sent: Liverpool_vs_BayernMunich|[RED] Direct red card for Musiala at 61 (Player expelled)
Sent: Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Salah) at 66' - Liverpool 6-1 BayernMunich
Sent: Liverpool_vs_BayernMunich|[SUB] Kane replaced by Choupo-Moting at 77'
Sent: Liverpool_vs_BayernMunich|[FT] 90' Full time - Liverpool 6-1 BayernMunich
[TCP] Connection closed.
```

Figure 16: Liverpool vs Bayern Munich

```
./publisher/out/publisher_tcp 127.0.0.1 5050 Barcelona_vs_PSG --demo
[TCP] Connected to 127.0.0.1:5050
Sent: Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Dembele) at 6' - Barcelona 0-1 PSG
Sent: Barcelona_vs_PSG|[GOAL] De Jong scores (assist: Pedri) at 12' - Barcelona 1-1 PSG
Sent: Barcelona_vs_PSG|[GOAL] Ter Stegen scores (assist: Lewandowski) at 19' - Barcelona 2-1 PSG
Sent: Barcelona_vs_PSG|[GOAL] Araujo scores (assist: Ter Stegen) at 22' - Barcelona 3-1 PSG
Sent: Barcelona_vs_PSG|[GOAL] Kounde scores (assist: Pedri) at 34' - Barcelona 4-1 PSG
Sent: Barcelona_vs_PSG|[GOAL] Kounde scores (assist: Pedri) at 37' - Barcelona 5-1 PSG
Sent: Barcelona_vs_PSG|[HT] 45' Halftime - Barcelona 5-1 PSG
Sent: Barcelona_vs_PSG|[CARD] Yellow card for Hakimi at 55'
Sent: Barcelona_vs_PSG|[RED] Direct red card for Vitiinha at 61 (Player expelled)
Sent: Barcelona_vs_PSG|[GOAL] Kounde scores (assist: Lewandowski) at 66' - Barcelona 6-1 PSG
Sent: Barcelona_vs_PSG|[SUB] Mbappe replaced by Kolo Muani at 77'
Sent: Barcelona_vs_PSG|[FT] 90' Full time - Barcelona 6-1 PSG
[TCP] Connection closed.
```

Figure 17: Barcelona vs PSG

En las capturas siguientes, se ven los mensajes recibidos del broker que fueron enviados por los publicadores:

```

> ./subscriber/out/subscriber_tcp 127.0.0.1 5050 RealMadrid_vs_Juventus Liverpool_vs_BayernMunich Barcelona_vs_PSG
[Subscriber-TCP] Connected to broker 127.0.0.1:5050
[Subscriber-TCP] Subscribed to topic: RealMadrid_vs_Juventus | Local port: 64376
[Subscriber-TCP] Subscribed to topic: Liverpool_vs_BayernMunich | Local port: 64376
[Subscriber-TCP] Subscribed to topic: Barcelona_vs_PSG | Local port: 64376
RealMadrid_vs_Juventus|[GOAL] Rabiot scores (assist: Locatelli) at 6' - RealMadrid 0-1 Juventus
Liverpool_vs_BayernMunich|[GOAL] Kimmich scores (assist: Sane) at 6' - Liverpool 0-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Dembele) at 6' - Barcelona 0-1 PSG
RealMadrid_vs_Juventus|[RED] Direct red card for Rudiger at 12 (Player expelled)
Liverpool_vs_BayernMunich|[GOAL] Szoboszlai scores (assist: Mac Allister) at 12' - Liverpool 1-1 BayernMunich
Barcelona_vs_PSG|[GOAL] De Jong scores (assist: Pedri) at 12' - Barcelona 1-1 PSG
RealMadrid_vs_Juventus|[GOAL] Vlahovic scores (assist: Szczesny) at 18' - RealMadrid 0-2 Juventus
Liverpool_vs_BayernMunich|[GOAL] Alisson scores (assist: Salah) at 19' - Liverpool 2-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Ter Stegen scores (assist: Lewandowski) at 19' - Barcelona 2-1 PSG
RealMadrid_vs_Juventus|[GOAL] McKennie scores (assist: Locatelli) at 23' - RealMadrid 0-3 Juventus
Liverpool_vs_BayernMunich|[GOAL] TAA scores (assist: Alisson) at 22' - Liverpool 3-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Araujo scores (assist: Ter Stegen) at 22' - Barcelona 3-1 PSG
RealMadrid_vs_Juventus|[GOAL] Rabiot scores (assist: Bremer) at 35' - RealMadrid 0-4 Juventus
Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Mac Allister) at 34' - Liverpool 4-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Kounde scores (assist: Pedri) at 34' - Barcelona 4-1 PSG
RealMadrid_vs_Juventus|[CARD] Yellow card for Valverde at 38'
Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Mac Allister) at 37' - Liverpool 5-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Kounde scores (assist: Pedri) at 37' - Barcelona 5-1 PSG
RealMadrid_vs_Juventus|[HT] 45' Halftime - RealMadrid 0-4 Juventus
Liverpool_vs_BayernMunich|[HT] 45' Halftime - Liverpool 5-1 BayernMunich
Barcelona_vs_PSG|[HT] 45' Halftime - Barcelona 5-1 PSG
RealMadrid_vs_Juventus|[CARD] Yellow card for McKennie at 54'
Liverpool_vs_BayernMunich|[CARD] Yellow card for Kimmich at 55'
Barcelona_vs_PSG|[CARD] Yellow card for Hakimi at 55'
RealMadrid_vs_Juventus|[RED] Direct red card for Valverde at 61 (Player expelled)
Liverpool_vs_BayernMunich|[RED] Direct red card for Musiala at 61 (Player expelled)
Barcelona_vs_PSG|[RED] Direct red card for Vitinha at 61 (Player expelled)
RealMadrid_vs_Juventus|[GOAL] Courtois scores (assist: Bellingham) at 65' - RealMadrid 1-4 Juventus
Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Salah) at 66' - Liverpool 6-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Kounde scores (assist: Lewandowski) at 66' - Barcelona 6-1 PSG
RealMadrid_vs_Juventus|[SUB] Vinicius replaced by Modric at 77'
Liverpool_vs_BayernMunich|[SUB] Kane replaced by Choupo-Moting at 77'
Barcelona_vs_PSG|[SUB] Mbappe replaced by Kolo Muani at 77'
RealMadrid_vs_Juventus|[FT] 90' Full time - RealMadrid 1-4 Juventus
Liverpool_vs_BayernMunich|[FT] 90' Full time - Liverpool 6-1 BayernMunich
Barcelona_vs_PSG|[FT] 90' Full time - Barcelona 6-1 PSG

```

Figure 18: Suscriptor 1

```

> ./subscriber/out/subscriber_tcp 127.0.0.1 5050 RealMadrid_vs_Juventus
[Subscriber-TCP] Connected to broker 127.0.0.1:5050
[Subscriber-TCP] Subscribed to topic: RealMadrid_vs_Juventus | Local port: 64377
RealMadrid_vs_Juventus|[GOAL] Rabiot scores (assist: Locatelli) at 6' - RealMadrid 0-1 Juventus
RealMadrid_vs_Juventus|[RED] Direct red card for Rudiger at 12 (Player expelled)
RealMadrid_vs_Juventus|[GOAL] Vlahovic scores (assist: Szczesny) at 18' - RealMadrid 0-2 Juventus
RealMadrid_vs_Juventus|[GOAL] McKennie scores (assist: Locatelli) at 23' - RealMadrid 0-3 Juventus
RealMadrid_vs_Juventus|[GOAL] Rabiot scores (assist: Bremer) at 35' - RealMadrid 0-4 Juventus
RealMadrid_vs_Juventus|[CARD] Yellow card for Valverde at 38'
RealMadrid_vs_Juventus|[HT] 45' Halftime - RealMadrid 0-4 Juventus
RealMadrid_vs_Juventus|[CARD] Yellow card for McKennie at 54'
RealMadrid_vs_Juventus|[RED] Direct red card for Valverde at 61 (Player expelled)
RealMadrid_vs_Juventus|[GOAL] Courtois scores (assist: Bellingham) at 65' - RealMadrid 1-4 Juventus
RealMadrid_vs_Juventus|[SUB] Vinicius replaced by Modric at 77'
RealMadrid_vs_Juventus|[FT] 90' Full time - RealMadrid 1-4 Juventus

```

Figure 19: Suscriptor 2

```

> ./subscriber/out/subscriber_tcp 127.0.0.1 5050 Liverpool_vs_BayernMunich
[Subscriber-TCP] Connected to broker 127.0.0.1:5050
[Subscriber-TCP] Subscribed to topic: Liverpool_vs_BayernMunich | Local port: 64378
Liverpool_vs_BayernMunich|[GOAL] Kimmich scores (assist: Sane) at 6' - Liverpool 0-1 BayernMunich
Liverpool_vs_BayernMunich|[GOAL] Szoboszlai scores (assist: Mac Allister) at 12' - Liverpool 1-1 BayernMunich
Liverpool_vs_BayernMunich|[GOAL] Alisson scores (assist: Salah) at 19' - Liverpool 2-1 BayernMunich
Liverpool_vs_BayernMunich|[GOAL] TAA scores (assist: Alisson) at 22' - Liverpool 3-1 BayernMunich
Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Mac Allister) at 34' - Liverpool 4-1 BayernMunich
Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Mac Allister) at 37' - Liverpool 5-1 BayernMunich
Liverpool_vs_BayernMunich|[HT] 45' Halftime - Liverpool 5-1 BayernMunich
Liverpool_vs_BayernMunich|[RED] Direct red card for Kimmich at 55'
Liverpool_vs_BayernMunich|[RED] Direct red card for Musiala at 61 (Player expelled)
Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Salah) at 66' - Liverpool 6-1 BayernMunich
Liverpool_vs_BayernMunich|[SUB] Kane replaced by Choupo-Moting at 77'
Liverpool_vs_BayernMunich|[FT] 90' Full time - Liverpool 6-1 BayernMunich

```

Figure 20: Suscriptor 3

```

> ./subscriber/out/subscriber_tcp 127.0.0.1 5050 Barcelona_vs_PSG Liverpool_vs_BayernMunich
[Subscriber-TCP] Connected to broker 127.0.0.1:5050
[Subscriber-TCP] Subscribed to topic: Barcelona_vs_PSG | Local port: 64379
[Subscriber-TCP] Subscribed to topic: Liverpool_vs_BayernMunich | Local port: 64379
Liverpool_vs_BayernMunich|[GOAL] Kimmich scores (assist: Sane) at 6' - Liverpool 0-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Dembele) at 6' - Barcelona 0-1 PSG
Liverpool_vs_BayernMunich|[GOAL] Szoboszlai scores (assist: Mac Allister) at 12' - Liverpool 1-1 BayernMunich
Barcelona_vs_PSG|[GOAL] De Jong scores (assist: Pedri) at 12' - Barcelona 1-1 PSG
Liverpool_vs_BayernMunich|[GOAL] Alisson scores (assist: Salah) at 19' - Liverpool 2-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Ter Stegen scores (assist: Lewandowski) at 19' - Barcelona 2-1 PSG
Liverpool_vs_BayernMunich|[GOAL] TAA scores (assist: Alisson) at 22' - Liverpool 3-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Araujo scores (assist: Ter Stegen) at 22' - Barcelona 3-1 PSG
Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Mac Allister) at 34' - Liverpool 4-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Kounde scores (assist: Pedri) at 34' - Barcelona 4-1 PSG
Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Mac Allister) at 37' - Liverpool 5-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Kounde scores (assist: Pedri) at 37' - Barcelona 5-1 PSG
Liverpool_vs_BayernMunich|[HT] 45' Halftime - Liverpool 5-1 BayernMunich
Barcelona_vs_PSG|[HT] 45' Halftime - Barcelona 5-1 PSG
Liverpool_vs_BayernMunich|[CARD] Yellow card for Kimmich at 55'
Barcelona_vs_PSG|[CARD] Yellow card for Hakimi at 55'
Liverpool_vs_BayernMunich|[RED] Direct red card for Musiala at 61 (Player expelled)
Barcelona_vs_PSG|[RED] Direct red card for Vitinha at 61 (Player expelled)
Liverpool_vs_BayernMunich|[GOAL] Van Dijk scores (assist: Salah) at 66' - Liverpool 6-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Kounde scores (assist: Lewandowski) at 66' - Barcelona 6-1 PSG
Liverpool_vs_BayernMunich|[SUB] Kane replaced by Choupo-Moting at 77'
Barcelona_vs_PSG|[SUB] Mbappe replaced by Kolo Muani at 77'
Liverpool_vs_BayernMunich|[FT] 90' Full time - Liverpool 6-1 BayernMunich
Barcelona_vs_PSG|[FT] 90' Full time - Barcelona 6-1 PSG

```

Figure 21: Suscriptor 4

2.2 UDP

TCP es el protocolo que tiene envío de sin handshake, es más rápido que TCP pero no es confiable ya que algunos mensajes podrían no llegar o llegar fuera de orden. El flujo que sigue nuestra aplicación se describe a continuación:

1. El broker crea un socket UDP y escucha en un puerto esperando conexiones de publicadores y suscriptores.
2. Los suscriptores crean un socket UDP y se vincula a un puerto aleatorio.
3. El suscriptor manda un mensaje al broker que contiene las suscripciones que quiere.
4. Los publicadores crean un socket UDP (no hay necesidad de un handshake).
5. Los publicadores envían eventos de partidos al broker.
6. El broker busca todos los suscriptores de los partidos según los eventos que llegan.
7. El broker envía el evento a cada IP/puerto que corresponde.
8. El suscriptor recibe y muestra el mensaje de cada evento.

Primero mostramos como el suscriptor 1 manda mensajes al broker indicando a cuales partidos se va a suscribir:

udp						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	UDP	65	49622 → 5050 Len=33
2	0.000036	127.0.0.1	127.0.0.1	UDP	68	49622 → 5050 Len=36
3	0.000050	127.0.0.1	127.0.0.1	UDP	59	49622 → 5050 Len=27

Figure 22: Suscriptor 1 enviando mensajes a broker en UDP

En la próxima image, mostramos una captura del terminal del suscriptor indicando que ha enviado mensajes al broker:

```

> ./subscriber/out/subscriber_udp 127.0.0.1 5050 RealMadrid_vs_Juventus Liverpool_vs_BayernMunich Barcelona_vs_PSG
[Subscriber-UDP] Subscribed to topic: RealMadrid_vs_Juventus | Local port: 49622
[Subscriber-UDP] Subscribed to topic: Liverpool_vs_BayernMunich | Local port: 49622
[Subscriber-UDP] Subscribed to topic: Barcelona_vs_PSG | Local port: 49622

```

Figure 23: Suscriptor 1 enviando mensajes a broker en UDP (terminal)

```

> ./broker/out/broker_udp 5050
[Broker-UDP] Listening on port 5050...
[Broker] Created new topic #0: RealMadrid_vs_Juventus
[Broker-UDP] Subscriber 127.0.0.1:49622 subscribed to 'RealMadrid_vs_Juventus' (Topic #0, 1 total subs)
[Broker] Created new topic #1: Liverpool_vs_BayernMunich
[Broker-UDP] Subscriber 127.0.0.1:49622 subscribed to 'Liverpool_vs_BayernMunich' (Topic #1, 1 total subs)
[Broker] Created new topic #2: Barcelona_vs_PSG
[Broker-UDP] Subscriber 127.0.0.1:49622 subscribed to 'Barcelona_vs_PSG' (Topic #2, 1 total subs)

```

Figure 24: Suscriptor 1 en los logs del broker

UDP 49622 → 5050

02 00 00 00 45 00 00 3d	32 6a 00 00 40 11 00 00	... E.. = 2j..@...
7f 00 00 01 7f 00 00 01	c1 d6 13 ba 00 29 fe 3c).. SUBSCRIB E RealMa
53 55 42 53 43 52 49 42	45 7c 52 65 61 6c 4d 61	drid_vs_ Juventus
64 72 69 64 5f 76 73 5f	4a 75 76 65 6e 74 75 73	0a

02 00 00 00 45 00 00 40	36 e9 00 00 40 11 00 00	... E.. @ 6 .. @...
7f 00 00 01 7f 00 00 01	c1 d6 13 ba 00 2c fe 3f ,..? SUBSCRIB E Liverp
53 55 42 53 43 52 49 42	45 7c 4c 69 76 65 72 70	ool_vs_B ayernMun
6f 6f 6c 5f 76 73 5f 42	61 79 65 72 6e 4d 75 6e	69 63 68 0a

02 00 00 00 45 00 00 37	bf 7f 00 00 40 11 00 00	... E.. 7 .. @...
7f 00 00 01 7f 00 00 01	c1 d6 13 ba 00 23 fe 36 #.6
53 55 42 53 43 52 49 42	45 7c 42 61 72 63 65 6c	SUBSCRIB E Barcel
6f 6e 61 5f 76 73 5f 50	53 47 0a	ona_vs_P SG.

Figure 25: Suscripciones enviadas al broker por suscripcion 1

En el siguiente screenshot se ven los datos que los publicadores han enviado al broker y se han despachado a los suscriptores:

21 94.815516	127.0.0.1	127.0.0.1	UDP	93 52519 → 5050 Len=61
22 94.815656	127.0.0.1	127.0.0.1	UDP	93 5050 → 49622 Len=61
23 94.815671	127.0.0.1	127.0.0.1	UDP	93 5050 → 49875 Len=61
24 95.143364	127.0.0.1	127.0.0.1	UDP	92 58210 → 5050 Len=60
25 95.143452	127.0.0.1	127.0.0.1	UDP	92 5050 → 49622 Len=60
26 95.143466	127.0.0.1	127.0.0.1	UDP	92 5050 → 5050 Len=60
27 95.143474	127.0.0.1	127.0.0.1	UDP	92 5050 → 55167 Len=60
28 95.818225	127.0.0.1	127.0.0.1	UDP	132 52519 → 5050 Len=100
29 95.818334	127.0.0.1	127.0.0.1	UDP	132 5050 → 49622 Len=100
30 95.818358	127.0.0.1	127.0.0.1	UDP	132 5050 → 49875 Len=100
31 96.148435	127.0.0.1	127.0.0.1	UDP	96 58210 → 5050 Len=64
32 96.148489	127.0.0.1	127.0.0.1	UDP	96 5050 → 49622 Len=64
33 96.148499	127.0.0.1	127.0.0.1	UDP	96 5050 → 50746 Len=64
34 96.148506	127.0.0.1	127.0.0.1	UDP	96 5050 → 55167 Len=64

sation completeness b bytes on wire (768 bits), 96 bytes captured (768 bits)
Completeness = 16% [Family: IP (2)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
User Datagram Protocol, Src Port: 58210, Dst Port: 5050
Data: 4:6976657278616f6c5f76735f42617965726e4d756e6963687c5b434152445d2059656c6c6f77206361726420666f7220
(Length: 64)

Figure 26: Mensajes de eventos de partidos UDP

Enseguida, estan las capturas completas de UDP:

Figure 27: Log de broker completo

```

> ./subscriber/out/subscriber_udp 127.0.0.1 5050 RealMadrid_vs_Juventus Liverpool_vs_BayernMunich Barcelona_vs_PSG
[Subscriber-UDP] Subscribed to topic: RealMadrid_vs_Juventus | Local port: 49622
[Subscriber-UDP] Subscribed to topic: Liverpool_vs_BayernMunich | Local port: 49622
[Subscriber-UDP] Subscribed to topic: Barcelona_vs_PSG | Local port: 49622
RealMadrid_vs_Juventus|[CARD] Yellow card for Szczesny at 6'
Liverpool_vs_BayernMunich|[CARD] Yellow card for Kane at 6'
RealMadrid_vs_Juventus|[GOAL] Courtois scores (assist: Carvajal) at 11' - RealMadrid 1-0 Juventus
Liverpool_vs_BayernMunich|[CARD] Yellow card for Kimmich at 11'
Barcelona_vs_PSG|[GOAL] De Jong scores (assist: Kounde) at 6' - Barcelona 1-0 PSG
RealMadrid_vs_Juventus|[CARD] Yellow card for Carvajal at 19'
Liverpool_vs_BayernMunich|[RED] Second yellow for Kane at 20' (Player expelled)
Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Marquinhos) at 11' - Barcelona 1-1 PSG
RealMadrid_vs_Juventus|[RED] Direct red card for Rudiger at 22 (Player expelled)
Liverpool_vs_BayernMunich|[CARD] Yellow card for Musiala at 27'
Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Marquinhos) at 15' - Barcelona 1-2 PSG
RealMadrid_vs_Juventus|[GOAL] Valverde scores (assist: Vinicius) at 32' - RealMadrid 2-0 Juventus
Liverpool_vs_BayernMunich|[CARD] Yellow card for Davies at 30'
Barcelona_vs_PSG|[GOAL] Donnarumma scores (assist: Hakimi) at 26' - Barcelona 1-3 PSG
RealMadrid_vs_Juventus|[GOAL] Rabiot scores (assist: Szczesny) at 40' - RealMadrid 2-1 Juventus
Liverpool_vs_BayernMunich|[GOAL] Musiala scores (assist: Kimmich) at 39' - Liverpool 0-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Ter Stegen scores (assist: Kounde) at 29' - Barcelona 2-3 PSG
RealMadrid_vs_Juventus|[HT] 45' Halftime - RealMadrid 2-1 Juventus
Liverpool_vs_BayernMunich|[HT] 45' Halftime - Liverpool 0-1 BayernMunich
Barcelona_vs_PSG|[CARD] Yellow card for Ter Stegen at 38'
RealMadrid_vs_Juventus|[GOAL] Bremer scores (assist: Rabiot) at 54' - RealMadrid 2-2 Juventus
Liverpool_vs_BayernMunich|[RED] Second yellow for Musiala at 56' (Player expelled)
Barcelona_vs_PSG|[HT] 45' Halftime - Barcelona 2-3 PSG
RealMadrid_vs_Juventus|[GOAL] Vinicius scores (assist: Courtois) at 59' - RealMadrid 3-2 Juventus
Liverpool_vs_BayernMunich|[CARD] Yellow card for Van Dijk at 59'
Barcelona_vs_PSG|[GOAL] Dembele scores (assist: Hakimi) at 51' - Barcelona 2-4 PSG
RealMadrid_vs_Juventus|[GOAL] Bellingham scores (assist: Courtois) at 66' - RealMadrid 4-2 Juventus
Liverpool_vs_BayernMunich|[SUB] Davies replaced by Laimer at 66'
Barcelona_vs_PSG|[GOAL] Marquinhos scores (assist: Donnarumma) at 59' - Barcelona 2-5 PSG
RealMadrid_vs_Juventus|[SUB] McKennie replaced by Kean at 76'
Liverpool_vs_BayernMunich|[GOAL] Neuer scores (assist: Kimmich) at 76' - Liverpool 0-2 BayernMunich
Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Mbappe) at 66' - Barcelona 2-6 PSG
RealMadrid_vs_Juventus|[FT] 90' Full time - RealMadrid 4-2 Juventus
Liverpool_vs_BayernMunich|[FT] 90' Full time - Liverpool 0-2 BayernMunich
Barcelona_vs_PSG|[SUB] Donnarumma replaced by Zaire-Emery at 76'
Barcelona_vs_PSG|[FT] 90' Full time - Barcelona 2-6 PSG

```

Figure 28: Log de Suscriptor 1

```

> ./subscriber/out/subscriber_udp 127.0.0.1 5050 RealMadrid_vs_Juventus
[Subscriber-UDP] Subscribed to topic: RealMadrid_vs_Juventus | Local port: 49875
RealMadrid_vs_Juventus|[CARD] Yellow card for Szczesny at 6'
RealMadrid_vs_Juventus|[GOAL] Courtois scores (assist: Carvajal) at 11' - RealMadrid 1-0 Juventus
RealMadrid_vs_Juventus|[CARD] Yellow card for Carvajal at 19'
RealMadrid_vs_Juventus|[RED] Direct red card for Rudiger at 22 (Player expelled)
RealMadrid_vs_Juventus|[GOAL] Valverde scores (assist: Vinicius) at 32' - RealMadrid 2-0 Juventus
RealMadrid_vs_Juventus|[GOAL] Rabiot scores (assist: Szczesny) at 40' - RealMadrid 2-1 Juventus
RealMadrid_vs_Juventus|[HT] 45' Halftime - RealMadrid 2-1 Juventus
RealMadrid_vs_Juventus|[GOAL] Bremer scores (assist: Rabiot) at 54' - RealMadrid 2-2 Juventus
RealMadrid_vs_Juventus|[GOAL] Vinicius scores (assist: Courtois) at 59' - RealMadrid 3-2 Juventus
RealMadrid_vs_Juventus|[GOAL] Bellingham scores (assist: Courtois) at 66' - RealMadrid 4-2 Juventus
RealMadrid_vs_Juventus|[SUB] McKennie replaced by Kean at 76'
RealMadrid_vs_Juventus|[FT] 90' Full time - RealMadrid 4-2 Juventus

```

Figure 29: Log de Suscriptor 2

```

> ./subscriber/out/subscriber_udp 127.0.0.1 5050 Liverpool_vs_BayernMunich
[Subscriber-UDP] Subscribed to topic: Liverpool_vs_BayernMunich | Local port: 50746
Liverpool_vs_BayernMunich|[CARD] Yellow card for Kane at 6'
Liverpool_vs_BayernMunich|[CARD] Yellow card for Kimmich at 11'
Liverpool_vs_BayernMunich|[RED] Second yellow for Kane at 20' (Player expelled)
Liverpool_vs_BayernMunich|[CARD] Yellow card for Musiala at 27'
Liverpool_vs_BayernMunich|[CARD] Yellow card for Davies at 30'
Liverpool_vs_BayernMunich|[GOAL] Musiala scores (assist: Kimmich) at 39' - Liverpool 0-1 BayernMunich
Liverpool_vs_BayernMunich|[HT] 45' Halftime - Liverpool 0-1 BayernMunich
Liverpool_vs_BayernMunich|[RED] Second yellow for Musiala at 56' (Player expelled)
Liverpool_vs_BayernMunich|[CARD] Yellow card for Van Dijk at 59'
Liverpool_vs_BayernMunich|[SUB] Davies replaced by Laimer at 66'
Liverpool_vs_BayernMunich|[GOAL] Neuer scores (assist: Kimmich) at 76' - Liverpool 0-2 BayernMunich
Liverpool_vs_BayernMunich|[FT] 90' Full time - Liverpool 0-2 BayernMunich

```

Figure 30: Log de Suscriptor 3

```

> ./subscriber/out/subscriber_udp 127.0.0.1 5050 Barcelona_vs_PSG Liverpool_vs_BayernMunich
[Subscriber-UDP] Subscribed to topic: Barcelona_vs_PSG | Local port: 55167
[Subscriber-UDP] Subscribed to topic: Liverpool_vs_BayernMunich | Local port: 55167
Liverpool_vs_BayernMunich|[CARD] Yellow card for Kane at 6'
Liverpool_vs_BayernMunich|[CARD] Yellow card for Kimmich at 11'
Barcelona_vs_PSG|[GOAL] De Jong scores (assist: Kounde) at 6' - Barcelona 1-0 PSG
Liverpool_vs_BayernMunich|[RED] Second yellow for Kane at 20' (Player expelled)
Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Marquinhos) at 11' - Barcelona 1-1 PSG
Liverpool_vs_BayernMunich|[CARD] Yellow card for Musiala at 27'
Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Marquinhos) at 15' - Barcelona 1-2 PSG
Liverpool_vs_BayernMunich|[CARD] Yellow card for Davies at 30'
Barcelona_vs_PSG|[GOAL] Donnarumma scores (assist: Hakimi) at 26' - Barcelona 1-3 PSG
Liverpool_vs_BayernMunich|[GOAL] Musiala scores (assist: Kimmich) at 39' - Liverpool 0-1 BayernMunich
Barcelona_vs_PSG|[GOAL] Ter Stegen scores (assist: Kounde) at 29' - Barcelona 2-3 PSG
Liverpool_vs_BayernMunich|[HT] 45' Halftime - Liverpool 0-1 BayernMunich
Barcelona_vs_PSG|[CARD] Yellow card for Ter Stegen at 38'
Liverpool_vs_BayernMunich|[RED] Second yellow for Musiala at 56' (Player expelled)
Barcelona_vs_PSG|[HT] 45' Halftime - Barcelona 2-3 PSG
Liverpool_vs_BayernMunich|[CARD] Yellow card for Van Dijk at 59'
Barcelona_vs_PSG|[GOAL] Dembele scores (assist: Hakimi) at 51' - Barcelona 2-4 PSG
Liverpool_vs_BayernMunich|[SUB] Davies replaced by Laimer at 66'
Barcelona_vs_PSG|[GOAL] Marquinhos scores (assist: Donnarumma) at 59' - Barcelona 2-5 PSG
Liverpool_vs_BayernMunich|[GOAL] Neuer scores (assist: Kimmich) at 76' - Liverpool 0-2 BayernMunich
Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Mbappe) at 66' - Barcelona 2-6 PSG
Liverpool_vs_BayernMunich|[FT] 90' Full time - Liverpool 0-2 BayernMunich
Barcelona_vs_PSG|[SUB] Donnarumma replaced by Zaire-Emery at 76'
Barcelona_vs_PSG|[FT] 90' Full time - Barcelona 2-6 PSG

```

Figure 31: Log de Suscriptor 4

```

> ./publisher/out/publisher_udp 127.0.0.1 5050 RealMadrid_vs_Juventus --demo
[UDP] Sending to 127.0.0.1:5050
Sent: RealMadrid_vs_Juventus|[CARD] Yellow card for Szczesny at 6'
Sent: RealMadrid_vs_Juventus|[GOAL] Courtois scores (assist: Carvajal) at 11' - RealMadrid 1-0 Juventus
Sent: RealMadrid_vs_Juventus|[CARD] Yellow card for Carvajal at 19'
Sent: RealMadrid_vs_Juventus|[RED] Direct red card for Rudiger at 22 (Player expelled)
Sent: RealMadrid_vs_Juventus|[GOAL] Valverde scores (assist: Vinicius) at 32' - RealMadrid 2-0 Juventus
Sent: RealMadrid_vs_Juventus|[GOAL] Rabiot scores (assist: Szczesny) at 40' - RealMadrid 2-1 Juventus
Sent: RealMadrid_vs_Juventus|[HT] 45' Halftime - RealMadrid 2-1 Juventus
Sent: RealMadrid_vs_Juventus|[GOAL] Bremer scores (assist: Rabiot) at 54' - RealMadrid 2-2 Juventus
Sent: RealMadrid_vs_Juventus|[GOAL] Vinicius scores (assist: Courtois) at 59' - RealMadrid 3-2 Juventus
Sent: RealMadrid_vs_Juventus|[GOAL] Bellingham scores (assist: Courtois) at 66' - RealMadrid 4-2 Juventus
Sent: RealMadrid_vs_Juventus|[SUB] McKennie replaced by Kean at 76'
Sent: RealMadrid_vs_Juventus|[FT] 90' Full time - RealMadrid 4-2 Juventus
[UDP] Socket closed.

```

Figure 32: Eventos de publicador 1

```

> ./publisher/out/publisher_udp 127.0.0.1 5050 Liverpool_vs_BayernMunich --demo
[UDP] Sending to 127.0.0.1:5050
Sent: Liverpool_vs_BayernMunich|[CARD] Yellow card for Kane at 6'
Sent: Liverpool_vs_BayernMunich|[CARD] Yellow card for Kimmich at 11'
Sent: Liverpool_vs_BayernMunich|[RED] Second yellow for Kane at 20' (Player expelled)
Sent: Liverpool_vs_BayernMunich|[CARD] Yellow card for Musiala at 27'
Sent: Liverpool_vs_BayernMunich|[CARD] Yellow card for Davies at 30'
Sent: Liverpool_vs_BayernMunich|[GOAL] Musiala scores (assist: Kimmich) at 39' - Liverpool 0-1 BayernMunich
Sent: Liverpool_vs_BayernMunich|[HT] 45' Halftime - Liverpool 0-1 BayernMunich
Sent: Liverpool_vs_BayernMunich|[RED] Second yellow for Musiala at 56' (Player expelled)
Sent: Liverpool_vs_BayernMunich|[CARD] Yellow card for Van Dijk at 59'
Sent: Liverpool_vs_BayernMunich|[SUB] Davies replaced by Laimer at 66'
Sent: Liverpool_vs_BayernMunich|[GOAL] Neuer scores (assist: Kimmich) at 76' - Liverpool 0-2 BayernMunich
Sent: Liverpool_vs_BayernMunich|[FT] 90' Full time - Liverpool 0-2 BayernMunich
[UDP] Socket closed.

```

Figure 33: Eventos de publicador 2

```

> ./publisher/out/publisher_udp 127.0.0.1 5050 Barcelona_vs_PSG --demo
[UDP] Sending to 127.0.0.1:5050
Sent: Barcelona_vs_PSG|[GOAL] De Jong scores (assist: Kounde) at 6' - Barcelona 1-0 PSG
Sent: Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Marquinhos) at 11' - Barcelona 1-1 PSG
Sent: Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Marquinhos) at 15' - Barcelona 1-2 PSG
Sent: Barcelona_vs_PSG|[GOAL] Donnarumma scores (assist: Hakimi) at 26' - Barcelona 1-3 PSG
Sent: Barcelona_vs_PSG|[GOAL] Ter Stegen scores (assist: Kounde) at 29' - Barcelona 2-3 PSG
Sent: Barcelona_vs_PSG|[CARD] Yellow card for Ter Stegen at 38'
Sent: Barcelona_vs_PSG|[HT] 45' Halftime - Barcelona 2-3 PSG
Sent: Barcelona_vs_PSG|[GOAL] Dembele scores (assist: Hakimi) at 51' - Barcelona 2-4 PSG
Sent: Barcelona_vs_PSG|[GOAL] Marquinhos scores (assist: Donnarumma) at 59' - Barcelona 2-5 PSG
Sent: Barcelona_vs_PSG|[GOAL] Hakimi scores (assist: Mbappe) at 66' - Barcelona 2-6 PSG
Sent: Barcelona_vs_PSG|[SUB] Donnarumma replaced by Zaire-Emery at 76'
Sent: Barcelona_vs_PSG|[FT] 90' Full time - Barcelona 2-6 PSG
[UDP] Socket closed.

```

Figure 34: Eventos de publicador 3

3 Preguntas de análisis

4 Tabla comparativa

Protocolo	Total Paquetes	Total Bytes	Duración (s)	Paquetes/s	Bytes/s	Retransmisiones (%)
TCP	292	26,773	24.69	11.83	1,084.27	47.6
UDP	149	16,859	603.64	0.25	27.93	N/A

Tabla 1: Resultados experimentales del análisis comparativo entre los protocolos TCP y UDP.

5 Análisis de Resultados del Laboratorio

5.1 ¿Qué ocurriría si en lugar de dos publicadores hubiera cien partidos simultáneos?

Si se aumentara el número de publicadores a cien, el desempeño del *broker* dependería mucho del protocolo utilizado.

Con **TCP**, cada publicador necesita su propia conexión, por lo que el *broker* debe mantener información de estado, buffers y descriptores de archivo por cada una. Esto hace que el consumo de memoria y CPU crezca de forma proporcional al número de conexiones. Además, si una de ellas es lenta, puede afectar la velocidad general de entrega por los mecanismos de control de flujo.

En cambio, con **UDP** el *broker* no conserva estado por conexión. Solo recibe y procesa los datagramas, lo cual permite manejar un volumen mayor de emisores con menor carga en el sistema. Sin embargo, también hay un mayor riesgo de pérdida de paquetes si la red se satura.

En resumen, TCP ofrece mayor confiabilidad, pero UDP resulta más liviano y escalable cuando el número de publicadores es muy alto.

5.2 Si un gol se envía como mensaje y un suscriptor no lo recibe en UDP, ¿qué implicaciones tendría?

En UDP, si un mensaje se pierde, el suscriptor simplemente no lo recibe. Esto podría generar que algunos usuarios no vean actualizaciones importantes como un gol, provocando desincronización entre los diferentes clientes.

TCP maneja mejor este tipo de situaciones, ya que garantiza que todos los mensajes lleguen al destino, aunque sea con un pequeño retraso. En una aplicación en tiempo real, la pérdida de un mensaje importante puede ser más perjudicial que un ligero aumento de la latencia, por lo que TCP sería más apropiado en este caso.

5.3 Protocolo más adecuado para seguimiento en vivo de partidos

La elección del protocolo depende del tipo de información que se transmite. Para eventos críticos como goles o tarjetas, lo ideal es usar **TCP**, ya que asegura que el mensaje llegue completo y en orden. En cambio, para datos que se actualizan constantemente y que pueden tolerar alguna pérdida, como estadísticas o posiciones, **UDP** es más eficiente por su baja latencia.

En las pruebas del laboratorio se observó que TCP entregó todos los mensajes de forma ordenada, mientras que UDP tuvo algunas pérdidas y reordenamientos. Por lo tanto, una solución híbrida podría ser la más adecuada en un escenario real, aunque agrega complejidad a la capa de aplicación.

5.4 Comparación de overhead entre TCP y UDP

El protocolo TCP utiliza más cabeceras (20 bytes de IP más 20 bytes de TCP) que UDP (20 bytes de IP más 8 bytes de UDP). Esta diferencia, aunque parezca pequeña, puede ser significativa cuando los mensajes son cortos o muy frecuentes.

En términos de eficiencia, UDP transmite más información útil por cada paquete, pero TCP compensa su sobrecarga con la ventaja de garantizar la entrega y el orden de los datos.

5.5 Efecto de desorden en UDP y posibles soluciones

Cuando los mensajes llegan fuera de orden, como si primero se recibiera el marcador 2–1 y luego el 1–1, la experiencia del usuario se ve afectada y puede generar confusión.

Para evitarlo, la aplicación puede incluir un número de secuencia o una marca de tiempo en cada mensaje, y procesarlos solo si son más recientes que el último recibido. También se pueden implementar confirmaciones (*ACKs*), retransmisiones selectivas o pequeños buffers que reordenen los mensajes antes de mostrarlos. Basicamente, se termina implementando un protocolo más parecido a quic, con mayor cantidad de bits en el header del mensaje para facilitar su organización.

5.6 Efecto del número de suscriptores sobre el desempeño

Con TCP, cada suscriptor tiene su propia conexión con el *broker*, por lo que al aumentar el número de suscriptores, también crece la carga en memoria y CPU y el uso del buffer por cada una de las conexiones. Además, el mismo mensaje debe enviarse de manera individual a cada cliente, lo que multiplica el tráfico.

Con UDP, el envío se realiza mediante datagramas y no se mantiene una conexión por cliente. Si la red soporta *multicast*, se puede enviar una sola copia del mensaje a todos los suscriptores al mismo tiempo, lo que mejora notablemente la eficiencia.

Basicamente, y como lo hemos mencionado anteriormente, UDP resulta mucho más rápido y eficiente, pero no es confiable como TCP.

5.7 Comportamiento ante caída del broker

Si el *broker* se detiene, las consecuencias son diferentes según el protocolo:

Con **TCP**, las conexiones se cierran o marcan error (esto a causa de que dejara de recibir los ACK que debería dentro de las ventanas hasta que se cumpla ese timeout designado), y tanto publicadores como suscriptores pueden detectar rápidamente la desconexión. Sin embargo, los mensajes que no se hayan entregado se pierden si el sistema no guarda persistencia.

En cambio, con **UDP**, los publicadores seguirán enviando mensajes, pero estos se perderán sin que nadie lo note. Los clientes tampoco sabrán que el *broker* dejó de funcionar.

Por lo tanto, TCP facilita la detección de fallos, aunque ninguno de los dos protocolos garantiza la recuperación automática sin mecanismos adicionales.

5.8 Sincronización simultánea de actualizaciones críticas

Esto es algo que es casi imposible, pues la velocidad a la que le llegan los paquetes depende de muchas fuentes de delay (delay por distancia, posibles cuellos de botella en la red de cada suscriber, etc), no solo del protocolo o de la capacidad del servidor. Teniendo esto en cuenta, se tendría que, a nivel de la capa de aplicación ingresar lógica para que ya recibido un mensaje que espere a que sea cierta hora para asegurar que a todos se les mostre de manera simultánea.

Ahora bien, si miramos que protocolo es el que facilita la implementación enfocandonos en la capa de transporte, el protocolo a elegir sería UDP, ya que en comparación es mucho más rápido que TCP, llevando que esa sensación de sincronización simultánea sea más factible de conseguir (por lo menos en aquellos que sí reciben su notificación) razón también por la que este protocolo es el más utilizado en videojuegos en línea que son en tiempo real.

5.9 Uso de CPU y memoria en el broker

Como lo mencionamos anteriormente, con TCP, el *broker* debe mantener información de cada conexión, incluyendo buffers, estados de ventana y retransmisiones. Esto hace que el consumo de memoria crezca con el número de clientes.

Con UDP, no se guarda información de conexión, por lo que el uso de memoria es menor. La carga de CPU proviene principalmente de la cantidad de datagramas que se deben procesar por segundo. En general, UDP tiende a ser más liviano y eficiente cuando se manejan muchos clientes.

5.10 Diseño de un sistema real para millones de usuarios

Para una aplicación a gran escala, lo más adecuado sería combinar ambos protocolos.

En la comunicación entre publicadores y el *broker*, se puede usar **TCP o QUIC** para asegurar que los eventos lleguen correctamente. Para distribuir la información a los usuarios finales, podría emplearse **UDP multicast** o redes de distribución de contenido (CDN) con WebSockets o HTTP/2, que mantienen buena velocidad y escalabilidad.

De esta manera, se logra un equilibrio entre fiabilidad y rendimiento: TCP asegura que la información importante no se pierda, mientras que UDP permite alcanzar una gran cantidad de usuarios con baja latencia.

Hoy en dia es comun que las aplicaciones mas grandes, como los navegadores web o aplicaciones como facebook o X utilizan quic para las funciones para reducir la latencia en sus aplicaciones, y apoyan los procesos utilizando una fuerte logica de la capa de aplicacion.

References

- [1] Computer Networking, a top-down approach. James Kurose, Keith Ross. Addison-Wesley, 6th ed.