UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Informe de Proyecto

Protocolos y modelos de los juegos en línea en tiempo real

Fecha: 13/09/2019
Integrantes: Josué Sandoval
Néstor Retamal
Gustavo Venegas
Asignatura: Redes de computadores I (ELO-322)
Profesor: Agustín González

Resumen

Las redes de computadores hicieron posible la conectividad entre distintos equipos en lugares distintos, diferentes aplicaciones la usan para dar servicios como mensajería en texto o transmisión de contenido multimedia, este trabajo abordará el uso que se les da en los videojuegos, más específicamente para permitir jugar con otros sistemas en tiempo real, los cuales podrían estar en diferentes partes del mundo. Se revisarán protocolos y modelos de trabajo usados para solucionar problemas que puedan surgir. En la parte práctica se como trabaja una partida en conexión LAN (Local Area Network) del videojuego Minecraft.

Introducción

El camino de los videojuegos en línea comienza con una adaptación de un juego de mesa el cual fue llamado MUD (Multi-User Dungeon) creado en 1978 por estudiantes de la universidad de Essex en Inglaterra. No fue sino hasta mucho después cuando la existencia y desarrollo de Internet permitieron la masificación de juegos como Quake (1996), Starcraft (1998), Lineage (1998) o Counter-Strike (1999) los cuales funcionaban en tiempo real, es decir, todos los jugadores participan al mismo tiempo en lugar de tener que tomar turnos **[ref 1][ref 2]**. Para estos propósitos se explicará el uso que se hace de los protocolos que transmiten datos rápidamente, pudiendo tolerar la pérdida de algunos datos y otros mecanismos que evitan que la partida se ralentice.

Protocolos

Como es sabido en la capa de transporte existen 2 protocolos los cuales son los más usados a la hora de transferencia de datos en tiempo real, el protocolo TCP (Transmission Control Protocol) y el protocolo UDP (User Datagram Protocol) ambos con sus distintos puntos a favor y en contra.

En un inicio el protocolo TCP fue ampliamente usado en el área de los videojuegos dado a que en este entonces UDP todavía no estaba totalmente desarrollado **[ref 3]** además de que TCP poseía una buena fiabilidad a la hora de enviar paquetes sin pérdidas. Sin embargo, con el pasar del tiempo el protocolo UDP fue cada vez más implementado en la industria de los videojuegos.

UDP VS TCP

TCP: Un protocolo fiable, bastante usando cuando es esencial que no haya pérdida de paquetes, gracias a que posee control de flujo y de congestión se lograba una baja pérdida de datos **[ref 4]**, sin embargo, esto causa retransmisiones que conlleva un retardo el cual puede volverse muy molesto en aplicaciones en tiempo real como la mayoría de los videojuegos. Se usó en los inicios de los juegos en líneas debido a que no existía UDP y por lo mismo, no había videojuegos en tiempo real.

UDP: Un protocolo rápido ya que sólo envía paquetes sin asegurarse que sean recibidos **[ref 5]**, empleado para aplicaciones que requieren rapidez, como los videojuegos donde las acciones son inmediatas y la jugabilidad rápida es necesaria. Es bastante usado en la actualidad para los videojuegos, para reducir los problemas generados por la pérdida de paquetes, se crearon mecanismos como la predicción de cliente, el cual también evita que una conexión lenta de uno de los jugadores ralentice toda la partida **[ref 6]**.

Modelos

Existen bastantes tipos de modelos, pero nos centraremos en los 3 predominantes, los modelos cliente-servidor, Peer-to-Peer, modelos híbridos **[ref 7]**. los cuales cumplen con la función de brindar una experiencia de juego en tiempo real, pero de distintas maneras.

Cliente servidor: En este modelo un servidor envía la información correspondiente a la partida a todos los usuarios y estos actualizan su información en su sesión siempre dependiendo del servidor. Una gran cantidad de juegos en línea actuales utilizan este modelo algunos ejemplos son World of Warcraft y Hearthstone **[ref 8]**.

Este modelo es económicamente escalable ya que permite la fácil implementación de microtransacciones además de la venta de suscripciones, soporta una gran cantidad de usuarios y además el poseedor del servidor tiene control absoluto en la partida, aunque este modelo es bastante caro de implementar posee una gran cantidad de ventajas.

Peer-to-Peer: En este modelo, cada cliente corre su propia versión del juego y se caracteriza por no tener algún servidor central, todos los usuarios solicitan información a los demás para

actualizar su información del juego y aunque hoy en día no es tan común verlos, se siguen usando en juegos RTS (Real Time Strategy) como Starcraft o Age of Empires 2.
Sin duda alguna este es el modelo más barato de implementar dado a que no requiere un servidor central el cual gestione todas las conexiones del juego, además de que siempre es necesario tener conexión a internet (Muchos de estos modelos funcionan con conexiones LAN).

Modelos Híbridos: Estos modelos se han hecho populares durante la última década, son mezcla del cliente servidor y peer-to-peer donde los usuarios están tanto conectados entre sí como con un servidor central, donde los usuarios envían información tanto al servidor como a los nodos, cada usuario tiene su propia sesión de juego y el servidor funciona como mediador en la partida.

Gracias a la mediación del servidor se tiene un cierto grado de estabilidad económica gracias a las microtransacciones, no es tan caro de implementar como un modelo cliente servidor puro dado a que los servidores propios de los Modelos híbridos no hostear partidas, solo hacen de mediador, esto no hace que sean perfectos dado a que da a la posibilidad a los nodos "hosters" de realizar uso malicioso de las redes cortando flujos de datos a propósito para arruinar la experiencia de algunos usuarios **[ref 9]**.

Resultados de parte práctica

i.

Primeramente, nuestro usuario buscará al servidor y su ip con mensajes ARP y IGMPv3 dado a que ambos se encuentran en la misma subred, la figura 1 muestra la captura de wireshark de dichos paquetes.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	169.254.102.93	224.0.2.60	UDP	90	61193 → 4445 Len=48
2	0.795756	169.254.129.237	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.2.60 for any sources
3	0.803070	169.254.129.237	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.2.60 for any sources
4	1.500624	169.254.102.93	224.0.2.60	UDP	90	61193 → 4445 Len=48
5	2.372522	AsustekC_c3:1f:b7	Broadcast	ARP	42	Who has 169.254.102.93? Tell 169.254.129.237
6	2.372758	fe80::a0e3:ed41:ceb…	ff02::1:3	LLMNR	107	Standard query 0xe5fd PTR 93.102.254.169.in-addr.arpa
7	2.372904	169.254.129.237	224.0.0.252	LLMNR	87	Standard query 0xe5fd PTR 93.102.254.169.in-addr.arpa
8	2.374760	fe80::497a:5a6:c899…	fe80::a0e3:ed41:ceb…	LLMNR	163	Standard query response 0xe5fd PTR 93.102.254.169.in-addr.arpa PTR LAPTOP-6HVE2LRG
9	2.374765	CompalIn_10:d5:28	AsustekC_c3:1f:b7	ARP	60	169.254.102.93 is at 98:28:a6:10:d5:28
10	2.374839	169.254.129.237	169.254.102.93	NBNS	92	Name query NBSTAT *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>
11	2.376832	169.254.102.93	169.254.129.237	LLMNR	143	Standard query response 0xe5fd PTR 93.102.254.169.in-addr.arpa PTR LAPTOP-6HVE2LRG
12	2.376833	169.254.102.93	169.254.129.237	NBNS	253	Name query response NBSTAT
13	3.000222	169.254.102.93	224.0.2.60	UDP	90	61193 → 4445 Len=48
14	3.001084	169.254.129.237	224.0.0.22	IGMPv3	54	Membership Report / Leave group 224.0.2.60
15	3.302432	169.254.129.237	224.0.0.22	IGMPv3	54	Membership Report / Leave group 224.0.2.60

Figura 1

ii.

Luego de encontrar al servidor el usuario procederá a enviar un requerimiento para entrar en la partida (Mediante un handshaking) el cual puede ser aceptado o denegado por el servidor, se muestra en la figura 2 dicha captura:

38	19.506003	169.254.102.93	224.0.2.60	UDP	90	61193 → 4445 Len=48
39	19.702883	169.254.129.237	169.254.102.93	TCP	66	59504 → 51381 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
40	19.703639	169.254.102.93	169.254.129.237	TCP	66	51381 → 59504 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
41	19.703748	169.254.129.237	169.254.102.93	TCP	54	59504 → 51381 [ACK] Seq=1 Ack=1 Win=525568 Len=0
42	19.705518	169.254.129.237	169.254.102.93	TCP	76	59504 → 51381 [PSH, ACK] Seq=1 Ack=1 Win=525568 Len=22
43	19.705717	169.254.129.237	169.254.102.93	TCP	65	59504 → 51381 [PSH, ACK] Seq=23 Ack=1 Win=525568 Len=11
44	19.706217	169.254.102.93	169.254.129.237	TCP	60	51381 → 59504 [ACK] Seq=1 Ack=23 Win=525568 Len=0
45	19.706914	169.254.102.93	169.254.129.237	TCP	60	51381 → 59504 [ACK] Seq=1 Ack=34 Win=525568 Len=0
46	19.708289	169.254.102.93	169.254.129.237	TCP	227	51381 → 59504 [PSH, ACK] Seq=1 Ack=34 Win=525568 Len=173
47	19.748100	169.254.129.237	169.254.102.93	TCP	54	59504 → 51381 [ACK] Seq=34 Ack=174 Win=525312 Len=0
48	21.006005	169.254.102.93	224.0.2.60	UDP	90	61193 → 4445 Len=48

Figura 2

iii.

Tiempo después el requerimiento es aceptado y la conexión TCP establecida el servidor procede a enviar una copia del “mapa” para que los dos estén sincronizados (Figura 3):

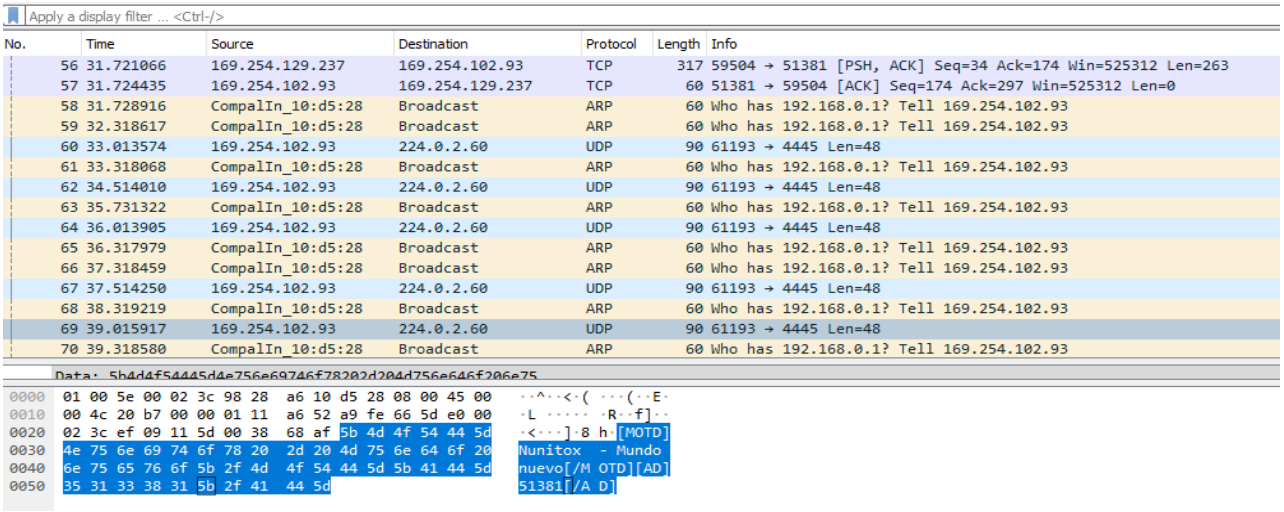
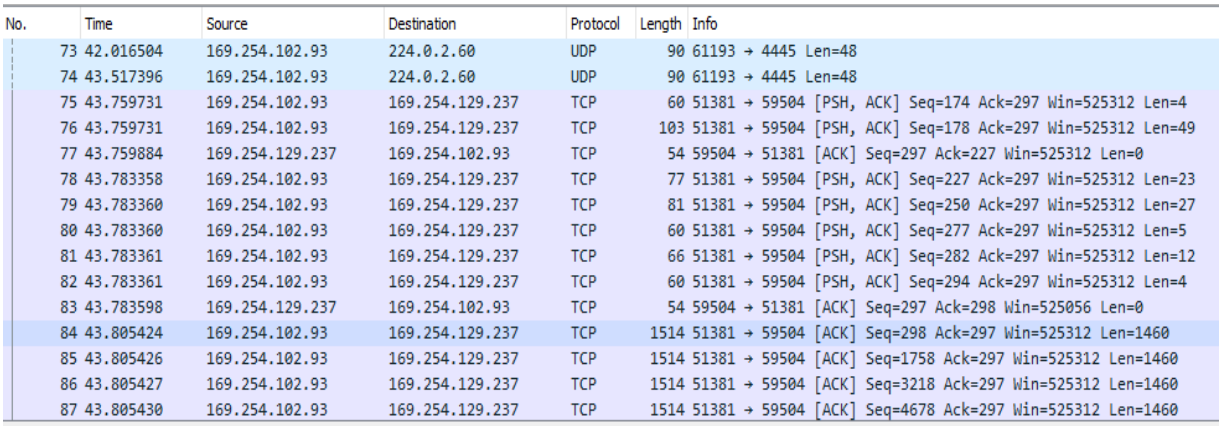


Figura 3

iv.

Finalmente, el usuario y el servidor se envían una gran cantidad de mensajes para actualizar sus sesiones de juego en tiempo real (Figura 4):

Figura 4



Conclusión.

En este trabajo analizamos superficialmente el funcionamiento de los videojuegos en tiempo real en el tema de las redes, analizando tanto los modelos implementados en estos como sus protocolos, por parte de los protocolos dimos a entender porqué antiguamente TCP era más usado que UDP y como tecnologías que hoy en día no son las más populares en algún futuro quizás puedan ser las bases de todo un sistema mucho más complejo.

Por parte de los modelos estudiamos los esquemas que impulsan la implementación de algunos modelos sobre otros, tanto si es que se quiere obtener un aprovechamiento económico de un producto mediante su interconectividad, como simplemente darle la capacidad de compartir sesiones simultáneas de juego o crear una experiencia interconectada pero intermediada por un tercero.

Sin duda alguna toda esta información nos podrá ser de ayuda si algún día en un futuro no muy lejano nos veamos en la necesidad de utilizar las herramientas de la interconectividad en un enfoque distinto, ya sea diseñando aplicaciones o mejorando lo que ya está escrito.

Referencias.

[1] https://en.wikipedia.org/wiki/Online_game **(Historia de los juegos en línea)**

[2] https://www.british-legends.com/CMS/index.php/about-mud1-bl/history **(Sitio oficial de MUD)**

[3] https://sites.google.com/site/lagranhistoriadelacomputacion/redes **(1980: creación del protocolo UDP)**

[4] https://tools.ietf.org/html/rfc793 **(Estandarización de TCP)**

[5] https://tools.ietf.org/html/rfc768 **(Estandarización de UDP)**

[6] https://en.wikipedia.org/wiki/Client-side_prediction

[7] https://es.wikipedia.org/wiki/Peer-to-peer **(Vease la parte de redes P2P hibridas).**

[8] http://elsemanal.es/desarrollo-de-juegos-en-red-parte-02-arquitectura-cliente-servidor

[9]https://www.mundogamers.com/noticia-lag-switch-y-los-tramposos-del-online.10982.html

Anexos

Paper sobre la predicción de cliente http://web.cs.wpi.edu/~claypool/courses/4513-B03/papers/games/bernier.pdf

Documento sobre el modelo Peer-to-Peer

http://www.vanderschaar-lab.com/papers/chapter_P2P_hpark.pdf

Documento sobre el modelo Cliente-Servidor

https://pdfs.semanticscholar.org/e1d2/133541a5d22d0ee60ee39a0fece970a4ddbf.pdf