Proyecto: Análisis de Protocolo de Red de FPS QuakeIII.

Vicente Inostroza
Patricio Navarrete
Eduardo Latorre

Universidad Técnica Federico Santa María
Departamento de Electrónica
Redes de Computadores

Agustín J. González V.

July 28, 2014

Abstract

En el presente documento se presenta la problemática de lograr una conexión rápida y fluida que permita conectar clientes de juego en un ambiente competitivo, y se analiza la solución implementada en el protocolo utilizado por el software Quake 3 Arena, consistente en un flujo de datos secuenciales transmitidos por UDP. Además, se resuelve el problema de la búsqueda de servidores por medio de un protocolo simple de consulta a un servidor maestro, y un sistema de handshaking para permitir la conexión con los mismos. Al final del trabajo se cuenta con una visión acabada de los detalles que hacen de este protocolo una solución robusta al problema planteado.

1 Introducción

1.1 Contexto

Como es sabido, *Internet*, en la década de los 90's fue una herramienta que revolucionó el mercado computacional , donde junto con éste también surgió con bastante fuerza los juegos multijugador, pues si bien por sí solos estos juegos de entretenimiento llamaban bastante la atención, el hecho de poder ocupar Internet para poder jugar con otros usuarios en tiempo real hizo crecer este mercado a tal punto de que difícilmente hoy en día encontramos juegos que no tienen la capacidad de multijugador.

Ante esto, surge la necesidad natural de poder comunicar a estos particulares clientes, con bajos tiempos de espera, con el fin de que la experiencia de juego en tiempo real pueda ser llevada satisfactoriamente, donde dicha necesidad es particularmente relevante en los juegos del tipo FPS (First Person Shooter).En base a lo mencionado, es que se desarrollará el cómo uno de los juegos más vendidos a finales de los 90; **Quake III Arena** , da solución a la necesidad básica mencionada de conexión en tiempo real.

1.2 Primeras aproximaciones.

1.2.1 Protocolo QuakeTest (1995)

Como primera aproximación, podemos mencionar el protocolo de una de las versiones previas del mencionado juego, donde destaca el uso de TCP en la capa de transporte y un gran tamaño de paquetes de datos. Esto trajo como consecuencia, en primer lugar, tiempos de procesamientos innecesarios al utilizar TCP, pues si bien es un protocolo bastante sofisticado para temas de confiabilidad, es una herramienta sobrestimada para los fines de una simple comunicación entre dos usuarios de un juego, donde además por otra parte el hecho de que los paquetes utilizados tengan un tamaño considerable (8 Kbytes), lleva a que sea necesario utilizar fragmentación y defragmentación de éstos para el traslado eficiente, considerando aún más que los enlaces de redes no destacaban por tener un ancho de banda adecuado en ese entonces.

1.2.2 Protocolo Quake2(1997)

Una segunda versión antecesora del protocolo a estudiar, es el protocolo Quake 2 , donde se utiliza directamente UDP como protocolo de capa de transporte, donde si bien esto mejoró el problema de tiempos de retardos, conllevó a tener diversos problemas de orden en el cual llegan los paquetes, aspecto fundamental a considerar en un tipo de juego FPS donde las acciones de forma consecutivas rigen el comportamiento natural de éstos.

1.2.3 Protocolo Quake 3 Arena

El protocolo a estudiar, considera toda la experiencia previa de sus antecesores, pues se basa en un enfoque completamente "no confiable" implementando UDP con el fin de minimizar tiempos de retardos, pero también

implementa la compresión de paquetes enviados, lo que elimina la fragmentación. Además considera, a nivel general, otras características a destacar como lo es la desactivación de la suma de comprobación.

Destacamos de ésto, de que si bien se está utilizando un protocolo no confiable, el protocolo a estudiar implementa a nivel de aplicación algunas características que le dan cierto grado de confiabilidad las cuales se detallarán más adelante.

2 Generalidades y Estructura de Datos

2.1 Estado de Juego

El sistema de juego se basa en un *estado de juego maestro*, el cual es modificado por medio de *comandos* ejecutados sobre el mismo por los distintos jugadores. Dichos comandos generan cambios sobre las propiedades del estado de juego maestro, los cuales son transmitidos al resto de los jugadores. El servidor lleva un registro de los cambios reconocidos por cada cliente, y sólo envía los cambios que no han sido reconocidos aún, y compila los mensajes a los distintos clientes basado únicamente en los cambios no reconocidos, reduciendo considerablemente el tamaño de los paquetes a enviar. Finalmente, para reducir todavía más el tamaño de los mismo, aplica compresión huffman basada en una tabla preestablecida incluida en el software.

2.2 Características del Protocolo

El protocolo Quake3 debe entregar los datos entre cliente y servidor de forma rápida y oportuna. En el contexto de los tiempos en FPS (First Person Shooters), esto significa que todo paquete que deba ser reenviado se consideraría ya obsoleto, por lo que un enfoque TCP generaría más latencia de la tolerable. Por lo tanto, el protocolo está basado en UDP, deshabilitando el uso del checksum para agilizar el procesamiento. Sin embargo, debido al esquema de estado de juego descrito anteriormente, el protocolo debe tener algún mecanismo para enviar los cambios de forma secuencia y de reconocer cuando se han aceptado cambios por parte del cliente. Por lo tanto, a nivel de aplicación el protocolo implementa números de secuencia y de acknowledge. Por lo demás, sólo existe un único tipo de paquetes, lo que agiliza aun más el procesamiento del mismo.

2.3 Qport

Para el caso donde más de un cliente se encuentra tras una misma IP y puerto, como el de estar tras un NAT dinámico, se debe implementar un mecanismo que permita distinguir al cliente destino tras el NAT y mantener la conexión aun si ocurren cambios de puerto. Para ello, el cliente genera en el momento de la conexión con el servidor un identificador el cual es incluido en todos los paquetes posteriores. Dicho identificador de 16 bits se denomina Qport.

2.4 Estructura de un Paquete de Datos

De esta forma, el paquete enviado desde el servidor al cliente para actualizarlo, o bien del cliente al servidor para modificarlo, tiene la siguiente estructura:

- Número de Secuencia (2 Bytes)
- Número de Acknowledge (2 Bytes)
- Qport (2 Bytes)
- Comandos (Variable)

3 Análisis de Server-Discovery usado por QuakeIII protocol 43.

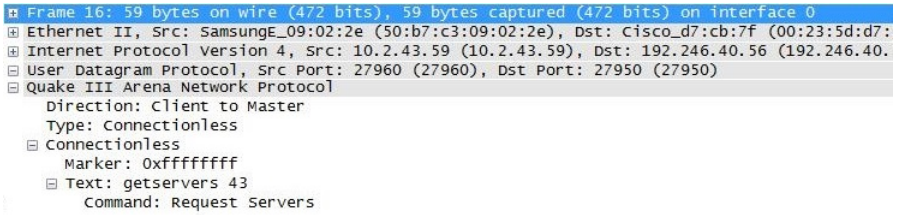
Esta sección se aborda brevemente la búsqueda del servidor de hosting utilizado por el protocolo QuakeIII.

En primera instancia se realiza un análisis de la secuencia de eventos que ocurren en la búsqueda de dicho servidor.

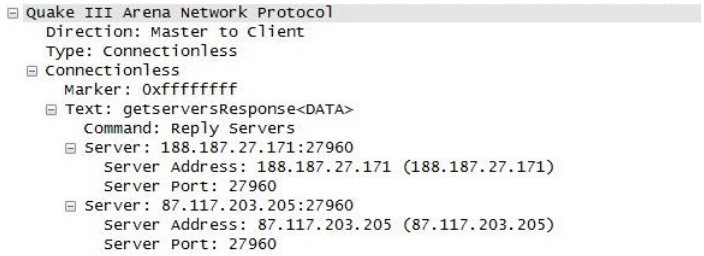
3.1 Secuencia de descubrimiento e inicio de partida.

Los servidores públicos de QuakeIII Arena se registran automáticamente al servidor maestro *master.quake3arena.com* en el puerto 27950. Por lo tanto este servidor centralizado posee la información de los servidores de juego que se encuentran disponibles alrededor del planeta.

- En primera instancia el cliente manda un paquete de requerimiento corto *getservers* al servidor maestro *master.quake3arena.com* al puerto 27950. Éste servidor central responde un paquete con una lista de todos los servidores de juego registrados actualmente. El servidor envía mensajes de respuesta múltiples en caso de que la lista de registrados sea demasiado larga para encajar en un sólo paquete UDP.



(a) Requerimiento simple *getservers*.



(b) Respuesta de servidor maestro, lista de servidores registrados.

Figure 1: Mensaje entre computador en red local y servidor maestro.

- Después de recibir la respuesta del servidor maestro, comienza una fase de pruebas del tipo $getinfo$ a los servidores de la lista en el mismo orden que lo envía el servidor maestro, con el fin de conocer aspectos relevantes de la partida, como el límite de jugadores, el tipo de juego, escenario, etc.
  - Ver anexo Paquete GETINFO.
- Cada servidor registrado informado por el servidor maestro, envía un paquete del tipo $inforesponse$ con información respecto al RTT, la aplicación despliega esta información en pantalla para dar la posibilidad al usuario de elegir el servidor con menor RTT.
  - Ver anexo INFORESPONSE.

  La siguiente imagen resume el intercambio de información entre el cliente, servidor maestro y servidores de juego disponibles

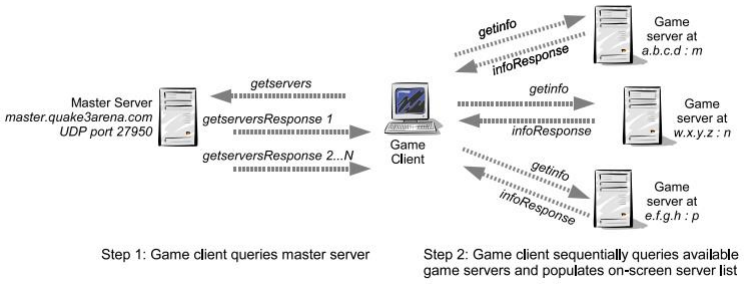


Figure 2: Server Discovery e $getinfo$.

- Luego de obtener la información de todos los servidores de juego disponibles, se elige uno y se comienza con la fase de Handshaking. Esta fase del protocolo se puede dividir en 2 partes. La primera contempla un requerimiento de desafío desde nuestra maquina hacia el servidor de juego, éste último responde con un código de partida, en este caso 715194606. Después, en la segunda fase, el cliente debe enviar todos los parámetros iniciales del personaje elegido como el modelo, el color, handicap, etc junto con el mismo código enviado por el servidor para unirse a la partida requerida anteriormente, mensaje al que el servidor responde con estado *conectado*, con lo cual comienza el intercambio de mensajes de juego, que será descrito mas adelante.

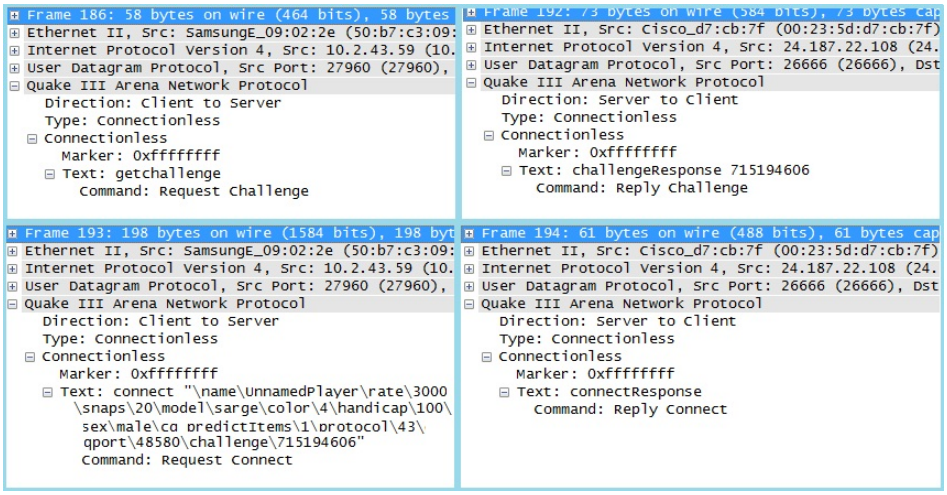


Figure 3: Captura de Handshaking, partida fuera de red local, protocolo $QuakeIII$.

4 Conclusiones

Cabe destacar que el avance tecnológico siempre nos está poniendo pruebas a superar, y del punto de vista de las redes de computadores no es la excepción, ante lo cual se plantean soluciones según las herramientas que se dispongan, que por cierto no son definitivas pues éstas se van optimizando, como fue el proceso de creación del protocolo estudiado, el cual ha sido referencia inclusive a la creación de otros protocolos de juegos más recientes y sofisticados.

Por otra parte podemos mencionar de que los conceptos aprendidos en el ramo de redes de computadores, nos son lo suficiente para poder entender ya como funciona un protocolo de red distinto a los estudiados en el ramo, validando ésto también con las mismas herramientas de estudio, como lo fue el caso de la utilización de *Wireshark* para el análisis de los paquetes de datos.

Un aspecto que queda en evidencia tras el estudio de este protocolo, es la integración de las características de distintos protocolos en una única solución que se adapta a los requerimientos de la aplicación en especifico, por lo tanto el desarrollo en la práctica no siempre sigue al pie de la letra los conceptos teóricos, como la separación de protocolos por capa.

References

[1] http://www.tilion.org.uk/2011/11/quake-3-network-format/

[2] http://trac.bookofhook.com/bookofhook/trac.cgi/wiki/Quake3Networking

[3] http://fabiensanglard.net/quake3/network.php

[4] CAIA Technical Report 070730A, July 2007, Server Discovery for Quake III Arena, Wolfenstein Enemy Territory and Quake 4

[5] CAIA Technical Report 110209A, February 2011, Quake III Arena Game Structures