Análisis del Desempeño de un Nuevo Algoritmo TCP Inalámbrico aplicado a un Escenario de Juegos

Alejandra Panchana, Román Lara-Cueva,Gonzalo Olmedo

***Resumen*—En este trabajo se presenta el estudio del comportamiento de un nuevo algoritmo TCP para redes inalámbricas sobre un escenario de juegos en tiempo real. Se realizaron pruebas con una topología punto-multipunto en dos escenarios que fueron una red interior, red exterior y una encuesta MOS. Los resultados, obtenidos a través del inyector de tráfico D-ITG determinaron el desempeño de la red y se elaboró un analisis cualitativo a través de una encuesta para establecer una medida subjetiva de la calidad percibida por el usuario en función de las diferentes condiciones de la red. Finalmente se presenta el análisis de los resultados obtenidos, donde los parámetros más considerados fueron el Jitter y Delay en los diferentes escenarios y las conclusiones de este proyecto.**

***Index Terms*—Protocolo TCP, delay, jitter, aplicación tiempo real, MOS.**

I. Introducción

En la actualidad se ha consolidado la tecnología Wi-Fi debido a su alta confiabilidad y beneficios que ofrece como su interoperabilidad con equipos de diferentes fabricantes, de manera que es una solución para extender la conectividad, junto al protocolo TCP existen diferentes propuestas [1] que permite mejorarlo modificando algunos parámetros, en [2] la medición de una red Wi-Fi en el cual mide el índice del rendimiento percibido por las conexiones TCP, donde los parámetros que afectan al rendimiento de manera directa es la ventana de congestión, la capa de enlace entre otros. En cuanto a los juegos de red, existe un incremento sustancial según [3], de manera que existe diferentes modelos y estudios realizados en donde determinan los parámetros principales que afectan al juego, tal como el retardo que existe en la red [4].

Además se han realizado diferentes estudios para determinar el desempeño del nuevo algoritmo TCP como se explica en [5], en ambientes *indoor* y *outdoor* con las diferentes tecnologías, por otro lado en [6] y [7] en una topología punto a punto se ha realizado estudios en las regiones costa e insular con la tecnología Wi-Fi a larga distancia. De esta manera se ha realizado diferentes estudios en base a la tecnología IEEE 802.11 como en [8], la cual han realizado pruebas de la tecnología WiLD en la región sierra, con el estándar 802.11b mediante dos saltos en cual alcanzo una distancia de 60km.

En el presente proyecto se tiene interés en probar un nuevo algoritmo TCP sobre un escenario de juegos en primera persona, donde fueron configurados dos jugadores y un servidor que están conectados de una manera inalámbrica a través de la tecnología IEEE 802.11 [9], de manera se estudiaron y midieron los par'ametros de calidad los cuales son *Delay*, *Jitter* y *Throughput* que son importantes en los juegos de red [10], para determinar el rendimiento de la red.

En base a lo mencionado, el contenido del paper está organizado de la siguiente manera: Sección II se presenta la metodología y materiales implementados. Sección III se describe el análisis de los resultados por último la Sección IV se presenta conclusiones y futuros trabajos.

II. Escenario de Pruebas

En esta sección se analizaron los materiales y métodos indispensables para el proceso de implementación del escenario de juegos entre el servidor y los dos clientes.

En los tres equipos utilizados que están conformado por dos hosts, y un servidor los cuales se implemento el Sistema Linux Ubuntu 10.04, debido a que es un Sistema Operativo de código abierto y es posible observar como se encuentra implementado el stack TCP/IP como se indica las modificaciones en [1], que contiene tres Kernels diferentes los cuales fueron modificados su ventana de contención por la propuesta en [10], que fue implementada de manera virtual en [11], como se ve en la Figura 1.

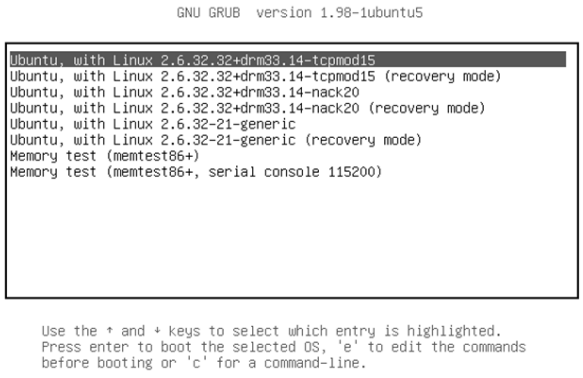

Fig. 1. Grub - Kernels

Para poder evaluar el nuevo algoritmo TCP en ambientes inalámbricos se utilizó un enlace punto-multipunto el cual es mostrado en la Figura 2, en el cual existe un *Access Point* (AP) que se comunica con varios puntos remotos, del escenario de juegos existe comunicación entre los remotos, los cuales pueden interactuar por medio del chat del servidor. Este tipo de topología presenta mayor eficiencia energética,

sincronización en la red y enrutamiento determinístico, pero no existe escalabilidad ni redundancia de caminos, lo que significa que no hay otro camino de respaldo al existir una falla en la red.

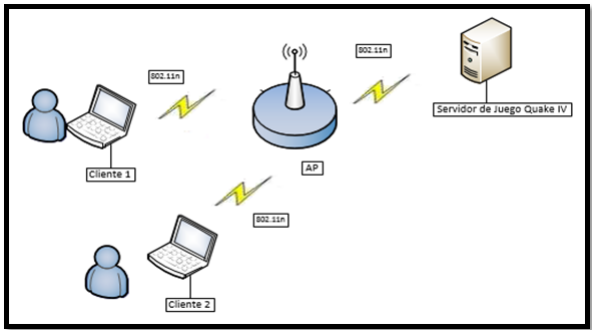

Fig. 2. Escenario de Pruebas

La transmisión de los paquetes de manera inalámbrica entre el servidor y los usuarios a través del protocolo TCP, por lo que se utilizó el software *Openvpn* [12] que trabaja sobre Sistema Operativo Linux debido a su flexibilidad, protección y fácil implementación de manera que esta permitió realizar cambios entre los protocolos.

Se introduce *Openvpn* que ofrece conectividad punto-a-punto con validación jerárquica de usuarios y host conectados remotamente, resulta una buena opción para tecnologías Wi-Fi debido que soporta una amplia configuración que utilizó una interfaz TUN que es una simulación de la capa de red, que permite conectar múltiples clientes remotos al servidor donde el cliente puede comunicarse con los otros clientes o no.

II-A. Configuración Cliente - Servidor

En el archivo *server.conf* creado al momento de instalar el software *OpenVPN*, y se configuraron los parámetros mostrados en la Cuadro I:

Cuadro I
Parámetros de Configuración

Parámetros	**Datos**
Modelo de Transmisión	TCP
Interfaz a Implementar	TUN
Importación de Claves	.crt, .key, .ca
Compresión	Dirección IP

Para poder hacer la transmisión de manera inalámbrica se utilizó el emulador TUN, el que permite para crear túneles virtuales que operan con el protocolo IP que se abren al momento de inicializar el *Openvpn*. De esta forma, se puede encapsular todos los paquetes que se transporten a través del tunel como *datagramas TCP*, en la capa 3. Finalmente, se realizaron pruebas de conectividad.

En los usuarios el proceso de configuración es similar que el servidor, pero en la dirección */etc/openvpn* se copian las claves de usuario creadas en el servidor y se edita el archivo *client.conf*, mostrado en la Figura 4, donde se configuró el cliente colocando la IP del servidor y el puerto en el cual va ingresar, sus respectivas claves y el protocolo a usar. Al final se reinicia el mismo programa hasta levantar la interfaz TUN. Como se observa en el Cuadro II.

Cuadro II
Parámetros de Configuración Cliente

Parámetros	**Datos**
Interfaz	dev tun
Claves	ca ca.crt cert client.crt key client.key
Dirección IP	remote 10,0,0,7 port 1194
Protocolo	tcp

Ajuste de Parámetros para el Escenario de Juegos

El juego implementado fue Quake IV, el mismo que se aloja en un servidor local que es usado por clientes para jugar contra el servidor o entre jugadores.

Servidor Quake IV

Para cumplir los requerimientos de la red en la implementación se consideró los permisos que requiere el juego para poder acceder a la interfaz gráfica que habilita 3D y se modificó el juego *dm.cfg* para poder asignar el tiempo en el cual se ejecutará, el número de jugadores, entre otras cosas tal como se ve en el Cuadro III. El número de jugadores depende si se encuentran *online* y puede alcanzar hasta 16 jugadores simultáneamente mientras que si esta *offline* solo alcanza máximo 8 jugadores [13].

Cuadro III
Archivo dm.cfg

Parámetros	**Datos**
Servidor Dedicado	1 = si, 0 = no
Máx. Jugadores	Hasta 8
Mín. de Jugadores	1
Paquetes para el juego	Q4dm1, Q4dm2, Q4dm3
Tiempo de juego [min]	1 - 60 min
Cargar tipo de juego	DM, Team DM, Tourney Arena CTF

Para ejecutar el archivo *dm.cfg* se coloca en la consola el comando *sudo ./quake4-dedicated +exec dm.cfg* en el path que se encuentra instalado, el cual presenta el menú del juego como se mira en el Cuadro IV.

Cuadro IV
Menú

Parámetros	Datos
begin game	1
player hit	0
player death	0
player kill	0
flag capture	0
end game	0

Cliente Quake IV

En el caso del cliente solo se ejecuta la linea de comando en terminal que es *sudo ./quake4-linux-1.4.2.x86.run*, la cual despliega el menú del juego, en el cual se ingresa a multijuagor para buscar el servidor de la red. Como se observa en la Figura 3.

Fig. 3. Interfaz del Juego

El cliente envía un dato llamado *heartbeat* cada intervalo de tiempo hacia el servidor, el cual contiene datos y procesos que están ocurriendo como el momento de que un jugador gana, pierde o se conecta al servidor.

Se implementó en dos escenarios, el primero fue en la red interior la cual tiene una distancia menor a 70m, mientras que el escenario exterior se realizo a una distancia alrededor de 160m.

Encuestas MOS

En esta sección se realizó la encuesta MOS que determina la Calidad del juego de manera subjetiva la cual describe la evacuación realizada por el jugador en base a la percepción general de un juego, ya que incluye cualquier factor que puede influir en el usuario de un juego. Un factor importante es la calidad sistema de juego como se muestra en la Figura 4, que describe los aspectos técnicos de la calidad del juego, por lo tanto se evalúa la calidad de la red y el objetivo de la calidad del juego [14].

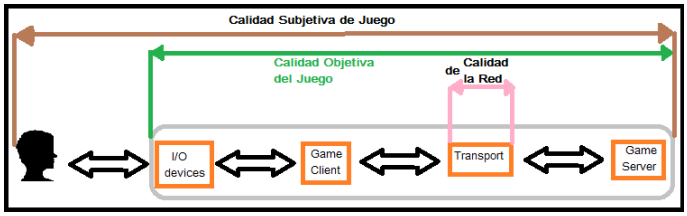

Fig. 4. Calidad del Juego

Con el fin de estimar la calidad que percibe el ser humano, se ha desarrollado cuatro preguntas en base al juego Quake VI cuyas preguntas fueron:

a) Observe atentamente las gráficas del juego y califique la calidad de vídeo.
b) Existió alguna anomalía en los movimientos del jugador?
c) El sonido corresponde al movimiento del jugador?
d) El movimiento del jugador obedece a los comandos que usted presiona a través del teclado?

III. Análisis de Resultados

En esta sección se analizan los resultados obtenidos en las pruebas, para los diferentes escenarios que se implementaron por lo que fueron obtenidos la cantidad de paquetes enviados, recibidos, perdidos o caídos, correspondientes al tráfico generado por el software D-ITG. Además fue fundamental examinar el *delay* y *jitter* ya que por medio de ellos se complementó los otros indicadores como la encuesta que se realizó mediante la opinión de los usuarios (MOS).

Como se observa en la Figura 5, la comunicación entre el servidor y el cliente se realizó una supervisión de tráfico por medio del *software Wireshark*, el cual permitió un análisis del protocolo de red. Para esta comunicación primero el transmisor envía un paquete SYN, el cual es el número de secuencia para crear la comunicación hacia el otro extremo, después en receptor envía datos como el ACK el cual indica que llegó el paquete, conjuntamente envía un paquete SYN el cual contiene el número de secuencia aleatorio así hasta concluir con la comunicación en el cual se envía un paquetes llamado FIN.

Se consideró la cantidad de paquetes que fueron entregados durante el enlace, ya que esto determinan el comportamiento que tiene la red. También en los resultados demuestran que no existen paquetes perdidos ya que el protocolo TCP hace que los paquetes perdidos se retransmitan hasta que lleguen a su destino. El tráfico fue capturado en un entorno inalámbrico siendo un salto de distancia entre el servidor y el cliente que su duración aproximada es de 2ms [14]. Por lo que la métrica escoge la mejor ruta, es decir el número de saltos lo que permite que la percepción del usuario con respecto al rendimiento de la red, sin embargo esto se ve afectado por la variación del retardo, *jitter* y ancho de banda.

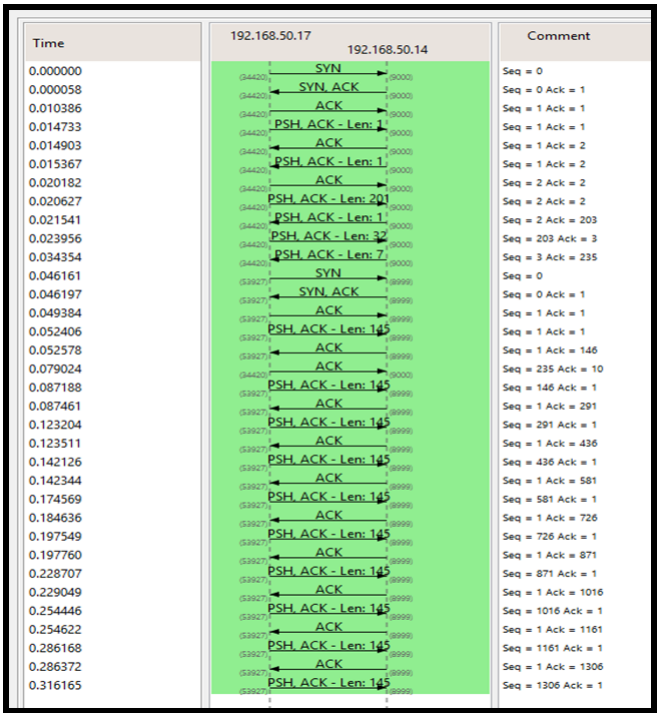

Fig. 5. Resumen del Proceso

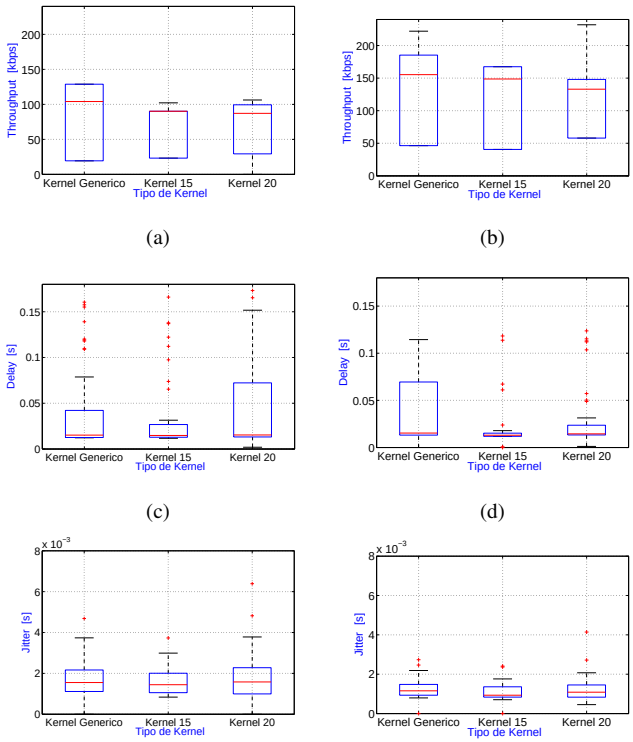

Fig. 6. Resultados **Throughput** (a) Uplink y (b) Downlink, **Delay** (c) Uplink y (d) Downlink, **Jitter** (e) Uplink y (f) Downlink

III-A. Escenario Interior

En el primer escenario se realizó en la red interiores donde se examinaron los datos del juego mediante el inyector de tráfico. En la Figura 6, se muestra las métricas obtenidas durante el análisis del escenario, en el cual se tiene seis paneles que corresponde al Uplink y Downlink de los parámetros.

III-A1. ***Throughput:*** Se define como una tasa de transferencia que permite analizar la velocidad de los datos que son transmitidos entre los terminales, está ligado con el ancho de banda del canal y el número de usuarios que comparten la red, si existen menos usuarios en la red el rendimiento del mismo sería más óptimo. Los resultados que presentan los niveles de *Throughput* fueron obtenidos con los diferentes paquetes. Como se observa en los paneles 6(a) y 6(b).

Cuadro V
Resultado de Throughput

Kernels	**Uplink Throughput** [$kbps$]	**Downlink Throughput** [$kbps$]
TCP Ventana 20	88,18	133,00
TCP Ventana 15	90,16	148,68
TCP Genérico	104,04	155,29

En el Cuadro V, se muestra el desempeño que se obtuvo del *Throughput*, y se observa que su valor menor es de 88,18 [$kbps$] en el enlace Uplink que pertenece al TCP20 con respecto al resto de Kernels, mientras que existe un mejor comportamiento en el enlace de Downlink con el TCP genérico con una tasa de transmisión de 155,29 [$kbps$] sin embargo los valores del enlace de subida y bajada en el Kernel 15 son aceptables.

III-A2. ***Delay:*** El *Delay* mide la cantidad del tiempo que tarda en viajar el paquete desde el origen al destino, por lo que esto es un factor muy importante para una aplicación en tiempo real, de manera que este se ve afectado por el tamaño y el número de paquetes. En los paneles 6(c) y 6(d), muestra el *Delay* que existe de extremo a extremo en tiempo de duración de 300s, los cuales se ven afectados por la distancia, tamaño y número de los paquetes.

Cuadro VI
Resultado de Delay

Kernels	**Uplink Delay** [ms]	**Downlink Delay** [ms]
TCP ventana 15	14,82	13,05
TCP ventana 20	15,36	15,51
TCP Genérico	15,15	15,30

En el Cuadro VI se presento menor retardo en Downlink es 14,82 [ms] y Uplink 13,05 [ms] que pertenece al Kernel 15, mientras que existe aumento en el resto de los Kernels como el valor mayor que es de 15,15 [ms], en cuanto al Uplink es

15,30 [ms] en el caso del Kernel Genérico. Por lo que se puede concluir que el Kernel modificado con ventana de congestión de 15 se presenta menores valores en tanto en Uplink como en Downlink.

III-A3. ***Jitter****:* En los paneles 6(e) y 6(f), *Jitter* se define como la variación en el tiempo en la llegada de los paquetes, causada por congestión de red o pérdida de sincronización para llegar al destino sin embargo si este valor es demasiado bajo se lo puede descartar pero este parámetro afecta mucho a la calidad del servicio cuando se tiene valores altos ya que afecta la calidad del flujo que puede ser variable debido a la congestión de tráfico de datos.

Cuadro VII
Resultado de Jitter

Kernels	**Uplink Jitter** [ms]	**Downlink Jitter** [ms]
TCP ventana 15	1,45	0,94
TCP ventana 20	1,66	1,09
TCP Genérico	1,56	1,17

En el Cuadro VII, se evidencia el comportamiento que el Kernel 15 tiene un valor menor en caso de Uplink que es de 1,45 [ms], mientras que Downlink el menor valor es 0,94 [ms], pero los kernels restantes se puede ver que existe un crecimiento como en el valor más alto que es 1,56[ms] y 1,17 [ms] de Kernel Genérico, pero su comportamiento es muy inestable cada vez que se aumentaba los paquetes al momento de la transmisiones. De manera que se puede reducir el problema del *jitter* mediante la utilización de un *buffer* de mayor conexión no obstante el aumento del *buffer* implica menos perdida de paquetes pero más retraso y una disminución implica menos retardo pero más pérdida de paquetes.

III-B. Escenario Exterior

En este escenario se evaluó el desempeño de la red inalámbrica implementada bajo el estándar *IEEE 802.11*, en el cual se analiza la calidad del enlace de acuerdo a varios parámetros obtenidos por medio de la herramienta D-ITG como *Throughput, Jitter y Delay*, se descarta los paquetes perdidos debido a que se implementó el protocolo TCP que proporciona un transporte fiable de flujo de bits entre las aplicaciones. En la Figura 7, se muestra las métricas obtenidas durante el análisis del escenario, en el cual se tiene seis paneles que corresponde al Uplink y Downlink de los parámetros.

III-B1. ***Throughput****:* Mediante los paneles 7(g) y 7(h), *Throughput* o rendimiento de la red de datos está dado por el número de bits que pueden ser transmitidos sobre la red en un cierto periodo de tiempo. Se realizaron medidas de *Throughput* en función del tiempo de la trama, y se utilizó la herramienta D-ITG con un tráfico de TCP. Es importante

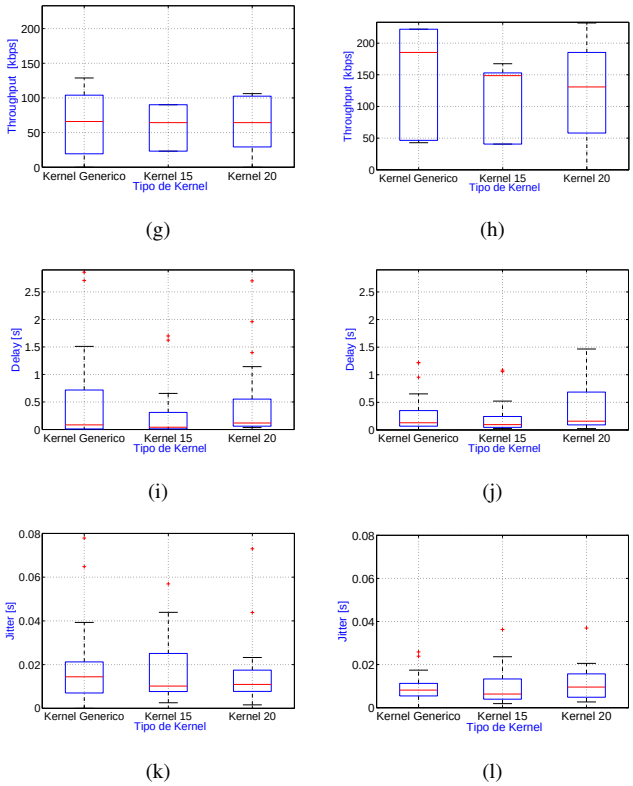

Fig. 7. Resultados **Throughput** (g) Uplink y (h) Downlink, **Delay** (i) Uplink y (j) Downlink, **Jitter** (k) Uplink y (l) Downlink

destacar que en las redes inalámbricas *IEEE 802.11*, la presencia de interferencias severas que producen la baja de transmisión de una manera drástica esto se debe a diferentes factores como los obstáculos, falta línea de vista y un factor importante es el uso del protocolo TCP en el cual hay que esperar el paquete ACK, este de vuelta por lo que esto afecta el rendimiento.

Cuadro VIII
Resultado de Throughput

Kernels]	**Uplink Troughput**[$kbps$]	**Downlink Troughput**[$kbps$]
TCP ventana 15	64,38	148,68
TCP ventana 20	63,90	94,41
TCP Genérico	82,56	185,28

En el Cuadro VIII, se muestra el *Throughput*, que fue obtenido después de realizar las pruebas con el inyector D-ITG durante un tiempo de 300 [s]. La inyección del tráfico corresponde al flujo de paquetes que se envían desde la dirección 192.168.20.18 del equipo emisor a la dirección destino 192.168.20.11, el flujo agregado entregó en el Kernel Genérico un resultado de 185,28 [kbit/s] y 82,56[kbit/s], de Uplink y Downlink respectivamente. En comparación con los valores del Kernel 15 como 64,38[kbit/s] y 148,68[kbit/s]. Pero el valor obtenido es aceptable para la aplicación en

tiempo real.

III-B2. ***Delay****:* Se determinó la calidad del servicio por medio de los parámetros que son medibles en la red como el retardo el cual el tiempo es la característica principal debido a que el paquete viaja desde el origen al destino. De tal forma el retardo es una comunicación entre las computadoras que incluye el tiempo de transporte en la red y el retardo de encolamiento del mismo. A continuación la desviación estándar es analizada en el escenario exterior por lo que es un promedio de las desviaciones individuales de los datos obtenidos durante el tiempo de simulación de 300 [s]. Como se observa en los panales 7(i) y 7(j).

Cuadro IX
Resultado de Delay

Kernels	**Uplink Delay** [ms]	**Downlink Delay** [ms]
TCP Ventana 15	74,38	10,17
TCP Ventana 20	117,96	10,94
TCP Genérico	159,02	15,50

En el Cuadro IX, muestra el *Delay*, que fue obtenido después de realizar las pruebas con el inyector D-ITG durante un tiempo de 300 segundos. La inyección del tráfico corresponde al flujo de paquetes que se envían desde la dirección 192.168.20.18 del equipo emisor a la dirección destino 192.168.20.11, el flujo agregado entregó como resultado un retardo con un valor mínimo de 74,38[ms] y 10,17[ms], de Uplink y Downlink respectivamente en el Kernel 15, no obstante se tuvo picos que tienen el valor de 159,02[ms] y 15,50 [ms] en el Kernel de ventana de congestión sin modificar en Uplink como Downlink ocasionados por el simulador.

III-B3. ***Jitter****:* El *Jitter* es considerado un parámetro muy importante dentro de la calidad de servicio, en caso de que sea excesivo es malo ya que la comunicación se ve afectada por lo que se tiene una comunicación entre cortada. En los panales 7(k) y 7(j), se puede observar que existe una diferencia de amplitud al momento de la transmisión de datos, ya que en los periodos en el cual se envía la información existe congestionamiento en la red, por causa de pérdida de sincronización o por las diferentes rutas seguidas por los paquetes para llegar al destino. De tal forma se observa en el Cuadro X, el menor valor es 10,18[ms] y 6,34[ms] que pertenece al Kernel 15, mientras que en el resto de los kernels con y sin modificación tiene un aumento del 75 % con respecto al Kernel 15.

Cuadro X
Resultado de Jitter

Kernels	**Uplink Jitter** [ms]	**Downlink Jitter** [ms]
TCP Ventana 15	10,18	6,34
TCP Ventana 20	10,94	9,23
TCP Genérico	15,50	8,56

De modo que los datos obtenidos se pudo evidenciar que el valor de *Jitter* fue mayor cuando se aumentó el flujo durante el tiempo de transmisión, esto se presenta cuando el tráfico es más denso y desordenado, provocando que los paquetes lleguen con retardos, ya que el transmisor como el receptor son diferentes, teniendo en cuenta que en algunos lapsos de tiempo no se encuentran sincronizados.

Finalmente, las *pérdidas de paquetes* son una de las principales causas del descenso de la calidad en redes pero en este proyecto se usó el protocolo TCP, que está orientado a conexión por lo que se comprobó que no existe paquetes perdidos durante el enlace, si esto sucediera por la congestión de la red dicho paquete se retransmite y se garantiza la información entre los extremos por lo que este paquete se considera como caído. Por lo que se puede concluir que en el protocolo TCP es una conexión de alta confiabilidad.De tal manera no existen paquetes perdidos, con un valor de 0 %, TCP realiza retransmisiones cuando existen paquetes caídos por lo que esto afecta claramente la eficiencia debido a que al momento de realizar una retransmisión el tamaño del paquete aumenta.

III-C. Encuesta MOS

Se determinó que la calidad del vídeo es 3 % mejor en el Kernel de ventana de congestión modificada 15 con respecto a los demás Kernel, mientras que existió una gran anomalía al momento de realizar los movimientos el jugador de 19 % en el Kernel de 20, en tanto que el Kernel de 15 obtuvo un valor del 9 %. El sonido corresponde al movimiento del jugador el 67 % de las personas encuestadas determinaron que el Kernel de 15 es mejor, sin embargo el Kernel de 20 es el que tuvo más problemas, por último el jugador obedece los comandos el 40 % considero que el óptimo es el Kernel de 15. Como se ve en el Cuadro XI

Cuadro XI
Resultado MOS

MOS	**Pregunta I**	**Pregunta II**	**Pregunta III**	**Pregunta IV**
TCP 15	67.70 %	5.14 %	67.18 %	40.31 %
TCP 20	64.15 %	19.11 %	52.08 %	35.38 %
TCP Genérico	65.15 %	14.71 %	54.68 %	32.61 %

IV. Conclusiones

En la red implementada Wi-Fi, se realizaron pruebas de conectividad, posteriormente se confirmó que la calidad del enlace es óptima se procedió a tomar los datos, que sirvieron para analizar el desempeño de la red. Se utilizó el software D-ITG para realizar las pruebas de inyección de tráfico con el objetivo de analizar el desempeño de la red, mediante los parámetros de calidad de servicio como *Delay*, *Jitter* y *Throughput*.

El *Jitter* en los tres escenarios implementados se determinó que el menor valor se encuentra que el Kernel de 15, por lo que existe una mejora con respecto al Kernel Genérico con un error del 6,49 %, aunque el Kernel de 20 tiene un error de 19 % por lo que no se ve una mejora importante, y esto afecta de manera significativa a la aplicación.

En cuanto al *Delay* se puede concluir que el Kernel de ventana de congestión 15 tiene un error de 2.18 % con respecto al Kernel Genérico por lo tanto lo hace óptimo para aplicación, sin embargo el Kernel de 20 tiene un error del 5,16 %, de manera que no es una buena opción implementar una aplicación en tiempo real ya que el *Delay* es uno de los parámetros más importantes de un juego.

En cuanto a los resultados obtenidos mediante las encuestas realizadas se determinó que el Kernel de la ventana de congestión 15 son aceptables para ser utilizado en una aplicación en tiempo real sobre una transmisión TCP debido a que la calidad del juego se ven afectada por diferentes factores como las gráficas del juego, movimiento del jugador, las capacidades físicas del cliente del juego como el sonido si existe aún retardo y por último la calidad de las redes.

En trabajo futuro se recomienda analizar el comportamiento de los buffer en función a su tamaño y también se podría ampliar la mezcla de tráfico e introduciendo mayor interactividad entre los jugadores tales como voz, salas de conversación y en dispositivos móviles.

V. Referencias

[1] Gonzalo Olmedo, Desempenho do Protocolo TCP em sistemas de comunicações sem fio CDMA usando estratégias de correção de erro FEC e RLP, en 2008.

[2] M. Franceschinis, M Mellia,"Measuring TCP over WiFi: A Real Case", en 2005.

[3] ISP planet news, http://www.ips-planet.com/news/20012/gamez_021202.html , en 2002

[4] J. Faerber, "Network game traffic modelling" ,en 2002

[5] Calvopiña Katherine, "Evaluación del Desempeño del algoritmo TCP modificado, en ambientes indoor y outdoor sobre las tecnologías Wi-Fi y WiMAX ".

[6] Cano & Almeida, "Análisis del desmepeño de una red con tecnlogía WI-FI para largas distnacias en la región costa del Ecuador",2012

[7] Sandoval & Acosta, "Evaluación del desempeño del protocolo TCP para un enlace inalámbrico de larga distancia inter-islas en la región insular",2014

[8] Barrionuevo & Tamayo, "Analisis del desempeño de una red con tecnología WiFi para largas distancias en un ambiente rural de la región Sierra",2011

[9] Saldaña, José, "Evaluación de la Calidad Subjetiva de Juegos Online según el Dispositivo de Acceso"

[10] Pablo Pila-Pais, "Verificacion del desempeño de un nuevo algoritmo de control de congestionamiento en entornos inalambricos reales mediante la modificacion del protocolo TCP en el Kernel de Linux" , en 2011

[11] Feilner, Markus, "OpenVPN: Building and integrating virtual private networks", en 2006

[12] Meyers, Philip,"Systems and methods for massively multi-player online role playing games",2004

[13] Schaefer, Christian and Enderes, "Subjective quality assessment for multiplayer real-time games", 2002

[14] Pozzobon, Mark, "Quake 3 packet and traffic characteristics ", en 2002