



Universidad
Zaragoza

Trabajo Fin de Máster

Herramienta de control de tráfico de red para el
análisis y evaluación de servicios multimedia
interactivos

Autor

David Giménez Muñoz

Directores

Julián Fernández-Navajas
Enrique F. Torres Moreno

Master Universitario en Ingeniería de Sistemas e Informática
Escuela de Ingeniería y Arquitectura
Diciembre, 2016



DECLARACIÓN DE
AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. David Giménez Muñoz,

con nº de DNI 17.740.530 - D en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo

de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la

Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)

Máster, (Título del Trabajo)

Herramienta de control de tráfico de red para el análisis y evaluación de
servicios multimedia interactivos.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada
debidamente.

Zaragoza, 25 de noviembre de 2016

Fdo: David Giménez Muñoz

Resumen

Para la realización de este trabajo se ha colaborado con el grupo de investigación de la Universidad de Zaragoza CeNITEQ (Communications Networks and Information Technologies For E-Health and Quality of Experience Group).

Un problema para los investigadores en redes, es el tiempo que tienen que invertir en configurar los entornos de experimentación. Tras analizar la metodología de trabajo del grupo, y trabajos relacionados en la materia de otros investigadores, se decidió implementar una herramienta que permita reproducir de manera automática, escalable y flexible, distintas condiciones en entornos controlados para diferentes tipos de experimentos que permitan analizar los casos estudiados.

Mastercraft, puede ser utilizada con una mínima instalación en cualquier sistema Linux. Permite crear distintos proyectos y dentro de estos, le muestra al usuario, de forma amigable, un menú con las distintas funciones que utilizar. Permitiendo capturar, procesar y modificar tráfico de red.

Realizar capturas de tráfico de red en el equipo local, pudiendo elegir el dispositivo de captura y múltiples capturas en equipos remotos para posteriormente almacenarlas para su tratamiento y procesado, filtrándolas y mostrando un resumen de los resultados de la captura de red.

También permite la modificación del tráfico de red de manera automática, remota y en cualquier momento, además de controlar las características de un determinado escenario de red permitiendo la repetitividad de los experimentos. Se pueden generar pérdidas de datos, retardos en el tráfico de la red, duplicar paquetes enviados, modificar los anchos de banda de las conexiones, etc

Para probar el buen funcionamiento de la herramienta se ha generado tráfico sintético con la herramienta Iperf y se realizan distintas capturas modificando el tráfico de red y viendo cómo cambia el comportamiento del medio.

Para probar y validar la utilidad de la herramienta en entornos de servicios multimedia interactivos, se utiliza Minecraft.

La herramienta es útil, procesa, captura y modifica el tráfico de red de la manera en la que se ha diseñado e implementado, es de fácil utilización, intuitiva y permite la automatización. Implementa el concepto de "Proyecto" que define su estructura.

Está publicada para su libre utilización y consulta en un repositorio "gitlab" de gestión de proyectos y control de versiones que posee la Universidad de Zaragoza, https://gitlab.unizar.es/dgimenez/Herramienta_Mastercraft .

Se ha creado un manual de usuario / programador.

La herramienta ha sido presentada en el congreso, II Workshop QoS y QoE en Comunicación Multimedia (QQCM) que se celebró en Zaragoza en el mes de Julio de 2016.

Índice

Contenido

Resumen	1
Índice	2
1. Introducción.....	4
1.1. Presentación	4
1.2. El problema.....	5
1.3. Posibles soluciones	6
1.4. Solución adoptada.....	7
1.5. Estructura de la memoria	9
2. Desarrollo de la herramienta.....	10
2.1. Aspectos generales de diseño e implementación	10
2.1.1. Proyecto	11
2.1.2. Bash	12
2.1.3. Estructura del Proyecto.....	15
2.2. Captura de tráfico de red.....	17
2.3. Procesado de datos	18
2.4. Modificación del tráfico de red.....	21
2.4.1. Preparar experimento.....	22
2.4.2. Generar experimento	23
2.4.3. Lanzar experimento	24
3. Validación y Casos de uso.....	25
3.1. Validación	25
3.1.1. Aspectos generales de validación	25
3.1.2. Captura y Procesado de datos.....	26
3.1.3. Modificación del tráfico de red	27
3.2. Casos de uso.....	28
3.2.1. Caso de uso 1: Modelado de tráfico	30
3.2.2. Caso de uso 2: Gestión de red	38
3.3. Conclusiones	44
4. Cronograma	45
5. Conclusiones.....	47
5.1. Conclusiones	47
5.2. Trabajos futuros	48
6. Bibliografía.....	50
Anexo 1. Estudio de los trabajos y metodología del grupo CeNITEQ.....	54

1. El grupo CeNITEQ.....	54
1.1. Estudio de los trabajos y metodologías.....	54
Anexo 2. Manual del Usuario.....	59
1. Instalación.....	59
2. Creación de un proyecto.....	60
3. Menú principal.....	62
4. La captura de tráfico de red.....	62
5. El procesado del tráfico.....	65
6. Compartir claves.....	69
7. Preparar experimento.....	72
8. Generar experimento.....	77
9. Lanzar experimento.....	79
10. Ejecutar comandos.....	80
11. Cambiar proyecto.....	81
Anexo 3 Tablas y Figuras.....	83

1. Introducción

1.1. Presentación

Cuando se creó la Word Wide Web el 12 de marzo de 1989 nadie creía que hoy en día todo estaría interconectado informáticamente, la tecnología avanza y cambia constantemente, y la investigación en redes de computadoras tiene que avanzar a la misma velocidad sino más, pues tiene que tener la vista puesta en nuevas redes de comunicaciones, nuevos escenarios globales y en las tecnologías que puedan hacer posible todos estos cambios.

Una red de computadoras u ordenadores, también llamada red informática o de comunicaciones de datos es un conjunto de equipos físicos informáticos conectados entre sí para transmitirse datos.

Los investigadores en redes trabajan los tres niveles de componentes que posee la red: software de aplicaciones, software de red y hardware de red. Para ello necesitan montar escenarios de pruebas para realizar sus experimentos.

En este ámbito, un entorno controlado de laboratorio es un escenario de pruebas donde se intenta reproducir el comportamiento de determinados elementos, como si estos estuvieran en el mundo real, o incluso determinar sus comportamientos reales.

En la siguiente imagen (Figura 1) se puede ver un ejemplo de cómo sería un típico escenario de pruebas.

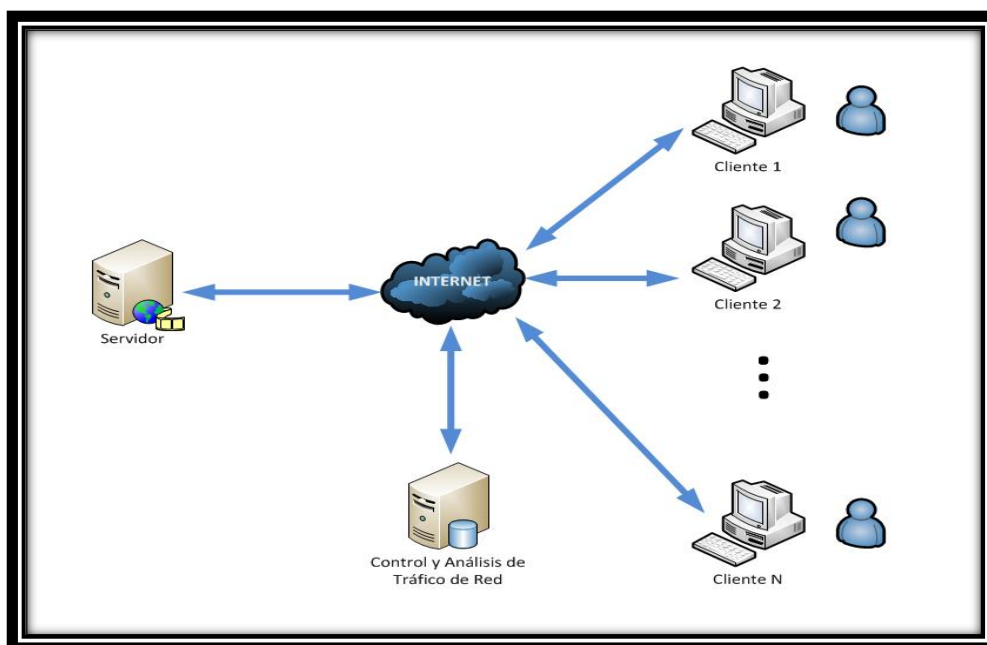


Figura 1 Escenario de pruebas

En este escenario se tendría un cliente o varios que hacen peticiones de datos a un servidor o servidores que contienen programas o aplicaciones. Estas peticiones pasarían por una red de comunicaciones con sus equipos de interconexión que serían

los que permiten la comunicación. Además se tendría un equipo o varios conectados a la red que se han llamado Control y Análisis de Tráfico de Red que monitorizarían dicho tráfico.

Uno de los grandes problemas en el ámbito de la investigación y más en particular en la universitaria, es la cantidad de tiempo que los investigadores invierten en la comprensión y manejo de distintas herramientas que les permiten realizar las diferentes configuraciones que requieren sus experimentos y su posterior tratamiento de los resultados de los mismos.

Para la realización de este trabajo fin de master, se ha colaborado con el grupo de investigación de la Universidad de Zaragoza CeNITEQ (Communications Networks and Information Technologies For E-Health and Quality of Experience Group) que está muy interesado en poder evaluar servicios multimedia interactivos en entornos de red (tanto del mundo real como de entornos controlados). Y así familiarizarse con las investigaciones del grupo de cara a realizar el presente trabajo.

Se necesita hacer un estudio en profundidad del grupo CeNITEQ para ver hacia qué estudios está enfocado, conocer los trabajos presentados y los estudios realizados, hablar con los investigadores y ver qué herramientas utilizan o necesitan para llevar a cabo técnicamente sus experimentos, y si dichas herramientas se pueden mejorar, actualizar o automatizar, para que sean más sencillas, más fáciles de usar y más intuitivas para el investigador, ver qué procedimientos se pueden mejorar o aglutinar y buscar soluciones que les puedan ofrecer. Para ello, el siguiente paso fue contactar con el grupo, estudiar los trabajos que realizan, seguir las investigaciones que realizan, tanto las suyas propias como las que realizan conjuntamente con otras instituciones y grupos a nivel internacional, analizar sus problemas y proponer posibles soluciones, las cuales se expondrán en el siguiente punto.

El estudio detallado sobre el grupo de investigación CeNITEQ, de sus métodos de trabajo y sus líneas de investigación, así como un resumen de los diversos estudios que los investigadores de este grupo realizan dentro del área de la telemática, como por ejemplo, las metodologías y herramientas para la mejora de las comunicaciones de red y la optimización de la calidad percibida, se encuentran en el anexo 1 de este trabajo.

1.2. El problema

Los investigadores pasan mucho tiempo aprendiendo el manejo de múltiples herramientas, como por ejemplo Tcpdump para la captura y procesamiento de datos, al igual que Tc y Netem para la gestión del tráfico de red, o el Shell de Linux para la ejecución de scripts y Matlab u Octave para el análisis de resultados. Estos se podrían considerar los cuatro pilares básicos donde se asientan o de donde parten muchas de sus investigaciones. Más adelante se explicarán Tcpdump, Tc, Netem, y otras, cómo las usan los investigadores y cómo se han utilizado para la construcción de la herramienta desarrollada.

Las investigaciones que se realizan dentro del campo de redes de ordenadores tienen que invertir mucho tiempo en el montaje de escenarios de red controlados, y en la comprensión de los parámetros de las herramientas utilizadas en estos.

Es muy común que haya que realizar múltiples experimentos teniendo que ejecutar dichas herramientas constantemente y que entre unos y otros experimentos se tengan que modificar los escenarios de pruebas o reconfigurar las herramientas antes de volver a ejecutarlas y la dedicación en tiempo que el investigador tiene que invertir es muy alta y al ser un trabajo manual la posibilidad de cometer errores humanos es muy elevada.

Además hay que asegurar que el experimento se pueda repetir en las mismas condiciones todas las veces que sean necesarias. Si dichas herramientas fueran más automáticas y su uso más sencillo, y los escenarios fueran más fácilmente reutilizables se ahorraría mucho tiempo.

Es común, en diversos entornos de investigación de redes de ordenadores, que los escenarios para la realización de las pruebas, tanto en entornos controlados de laboratorio, como en entornos reales como se ha visto en la Figura 1, puedan estar compuestos por:

- Distintas tecnologías de redes de comunicación
- Dispositivos de interconexión de redes (*router, switch, hub, etc*)
- Herramientas de monitorización de tráfico
- Aplicaciones controladoras o modificadoras de red
- Equipos clientes
- Equipos servidores

En este punto se hace muy interesante automatizar la generación de escenarios de prueba y en la medida de lo posible, que los procedimientos técnicos utilizados sean lo más sencillos y transparentes que se pueda para la labor diaria del investigador, haciendo que pueda dedicar el máximo de tiempo posible a su investigación y experimentos, y el mínimo tiempo posible a las configuraciones necesarias para llevar a cabo dichas labores.

Teniendo en cuenta todo lo anterior, el objetivo de este trabajo fin de master es analizar el procedimiento técnico, definir y crear una herramienta que automatice la realización de experimentos y su validación. Y como objetivo transversal este trabajo pretende ser una iniciación a la investigación.

1.3. Posibles soluciones

Entre las posibles soluciones para solventar el problema y conseguir nuestro objetivo de mejorar las actividades técnicas de los investigadores, estarían las cuatro siguientes:

- Contratación de personal técnico
- Uso de aplicaciones comerciales
- Hacer herramientas ad-hoc por parte de los Investigadores

- Creación de una macro-herramienta

La primera opción, contratar personal técnico especializado en programación de software orientado a entornos de red, resulta muy costosa y difícil de llevar a cabo. Uno de los motivos es la dificultad para encontrar a personal técnico especializado. Otro de estos motivos es el tema económico, y el burocrático. No se dispone de muchos de estos trabajadores y de los que se dispone, tienen otras muchas tareas en el grupo que les impiden dedicar un mínimo tiempo a este problema.

La segunda opción sería el uso de aplicaciones ya comercializadas que están actualmente en el mercado. Estas aplicaciones son propietarias y no permiten que su producto sea modificado de ninguna forma, con lo que dificultan los ajustes que se requieren para poder realizar los experimentos necesarios. Otro problema es que no están para todos los tipos de redes que se requieren como wifi, Ethernet, 3g, 4g o tecnologías aún no desarrolladas o en desarrollo, o están dentro de sistemas operativos propietarios que tampoco suelen permitir las modificaciones. Además suelen tener un precio elevado que puede ir incrementándose en el caso de que se tenga que pagar por licencias utilizadas o por módulos instalados. El problema es que se necesitarían muchas de estas herramientas para cubrir las necesidades del grupo, a causa de la imposibilidad de modificarlas para poder ajustarlas a los requisitos que se necesitan.

La tercera opción sería que cada investigador cree las herramientas que necesite, como ya han hecho con alguna, pero esta opción va completamente en contra de lo que ellos demandan y necesitan, que es poder dedicar el máximo tiempo posible a sus investigaciones.

La cuarta opción es la creación de una macro-herramienta que incorporara las herramientas que los investigadores ya utilizan, que fuera reutilizable y reconfigurable, automatizase los procesos repetitivos, como las conexiones remotas, que permitiera la organización por proyectos y fuera fácilmente ampliable. Un ejemplo de herramienta software creada por parte del grupo y que incluye alguno de los puntos anteriores es ETG [ES12], que es una herramienta para la automatización del análisis de tráfico, la captura y la generación de flujos IP multimedia que permitan obtener medidas de calidad en distintos entornos controlados de laboratorio, también en entornos reales empresariales, que sirva para la evaluación de distintas aplicaciones y tecnologías. Esta herramienta se centra únicamente en el análisis y la caracterización del tráfico sin entrar ni en gestión del tráfico ni en los otros aspectos dichos al principio del párrafo.

1.4. Solución adoptada

Después de hacer el estudio de los distintos procedimientos técnicos independientes unos de otros y generalmente utilizados de forma manual por parte de los investigadores y técnicos TI del grupo CeNITEQ , además de haber tenido distintas entrevistas con varios de sus investigadores y la constatación de la no existencia de ninguna herramienta que se ajustara a sus necesidades concretas, se pensó en la posibilidad de crear una nueva herramienta que aglutinara los procedimientos técnicos ya existentes, crease otros que fueran necesarios y los automatizara. Se pensó en desarrollarla, como trabajo previo para una posible investigación propia, realizarla además lo más genérica posible, para que el mayor número de investigadores, yo

incluido, pudiera utilizarla en todos sus experimentos dentro de los distintos proyectos que el grupo realiza.

A continuación y teniendo otra vez en cuenta los cuatro pilares básicos de los investigadores, que son captura y procesamiento de datos (Tcpdump), gestión de tráfico de red (Tc y Netem), ejecución de scripts (Shell de Linux) y análisis, se van a enumerar como funcionales (F) y no funcionales (NF) y también se van a definir, los requisitos que la herramienta tiene que tener:

- Usabilidad
 - (F1) Gestionar conjuntos de experimentos en lo que denominamos proyecto. Debe permitir crear distintos proyectos y dentro de estos, mostrarle al usuario, de la forma amigable, un menú con las distintas herramientas que tiene a su disposición para utilizar.
 - (F2) Tiene que mostrar al usuario tras la realización de cada experimento un resumen de los resultados de la captura de red, como por ejemplo, las IP que aparecen, el momento en el que aparecen por primera o última vez, su ancho de banda, los paquetes o bytes enviados o recibidos, tiempos de captura, etc. Esto permitiría al investigador rápidamente comprobar si el experimento no se ha realizado correctamente, cómo se ha comportado la red, pudiendo así repetir el procesado con otros parámetros, volver a realizar otra captura o acudir a los ficheros generados y sus gráficas y bucear en ellos de una manera más eficiente al estar estos datos ordenados por IP y tráfico enviado o recibido.
 - (NF1) La herramienta debe ser modular, que cada una de sus partes pueda ser usada de manera independiente.
 - (NF2) Modificable, que se puedan añadir aplicaciones o funciones que se necesiten más adelante.
 - (NF3) Fácil mantenimiento o actualización por parte de programadores investigadores.
 - (NF4) Qué sea portable y que no necesite instalación o que esta sea sencilla en Linux para facilitar su uso a los investigadores no programadores.
 - (NF5) Crear una guía para usuarios/programadores, para que se pueda acudir a ella para conocer cómo utilizar exactamente las distintas opciones o como sacarle el mayor partido a la aplicación más allá de sus opciones típicas.
- Captura y Procesamiento
 - (F3) Tiene que realizar capturas de tráfico de red pudiendo elegir el dispositivo de captura, almacenar dicha captura dentro del proyecto en la que ha sido ejecutada, para su posterior tratamiento ordenado.
 - (F4) Procesar las capturas filtrándolas por los distintos parámetros que se necesiten permitiendo para ello incluir expresiones complejas para facilitar su posterior análisis.
- Gestión de red
 - (F5) La herramienta debe permitir ser utilizada en cualquier sistema Linux, además varios de los programas utilizados por los investigadores integrados en la aplicación (Tc y Netem) solo funcionan en dicho sistema operativo y los

equipos de los laboratorios y de los investigadores o tienen arranque dual Windows/Linux o únicamente Linux.

- (F6) Igual que con el procesamiento, en la modificación del tráfico de red también debería permitir incluir expresiones complejas haciendo así mucho más potente la herramienta.
- (F7) Tiene que simplificar la configuración y la conexión automática a otros equipos remotos.
- (F8) Permitir la modificación del tráfico de red de manera automática, remota y en cualquier momento que se precise.
- (F9) Poder generar pérdidas de datos, retardos en el tráfico de la red, duplicación de paquetes enviados, modificar los anchos de banda de las conexiones y volver todo a su estado normal las veces que queramos y en cualquier momento.

1.5. Estructura de la memoria

Ya hemos visto la primera de las seis secciones de las que consta esta memoria, en la que se hace una introducción al trabajo, se analiza el problema planteado, sus posibles soluciones y la solución adoptada.

La segunda sección presenta la herramienta creada, mostrando cada uno de sus módulos y explicando detalladamente cómo se han cumplido los requisitos que tenía que cumplir la herramienta.

En la Tercera sección se presentan las distintas pruebas realizadas a la herramienta en distintos escenarios de uso.

La cuarta sección presenta las conclusiones del proyecto y los trabajos futuros y en la quinta sección se muestra la bibliografía.

Por último se incluyen varios anexos: en el primero se realiza un estudio sobre el grupo CeNITEQ sus trabajos y metodologías, en el segundo se muestra la guía de usuario de la herramienta realizada, y en el tercero y último se recopilan todas las tablas y gráficas generadas en los experimentos de la tercera sección.

2. Desarrollo de la herramienta

El objetivo principal de este trabajo fin de master como ya se comentó en el capítulo anterior, es analizar el procedimiento técnico, definir y crear una herramienta que automatice la realización de experimentos y su validación. Para llevar a cabo dicho objetivo, en este capítulo se presenta el diseño e implementación de la herramienta desarrollada para ello.

El capítulo se divide en cuatro puntos, aspectos generales del diseño y de la implementación, captura de tráfico de red, procesamiento de datos y modificación del tráfico de red. En cada uno de estos puntos y cogiendo como base la figura 2, donde se representa el diagrama general de la herramienta, explican cómo se realizó el diseño e implementación de la misma, qué programas se utilizaron, cómo se ha estructurado, etc.

Antes de comenzar con las decisiones de diseño, indicar que para cualquier aclaración o ampliación de información sobre la herramienta, "Mastercraft", así como para poder manejar o modificar la misma, se ha creado un Manual de usuario/programador que se encuentra para su consulta en el Anexo 2 de este trabajo.

La herramienta de control de tráfico de red para el análisis y evaluación de servicios multimedia interactivos, a la que he denominado Mastercraft, está publicada para su libre utilización y consulta en un repositorio "gitlab" de gestión de proyectos y control de versiones que posee la Universidad de Zaragoza, en la siguiente dirección: https://gitlab.unizar.es/dgimenez/Herramienta_Mastercraft, además existen copias en posesión del grupo de investigación CeNITEQ y de los directores de este trabajo fin de master.

2.1. Aspectos generales de diseño e implementación

Uno de los aspectos más importantes que hay que tener en cuenta a la hora de desarrollar la herramienta es que tiene que permitir a los investigadores reproducir, de manera escalable y flexible, distintas condiciones en entornos controlados para diferentes tipos de pruebas que permitan analizar los casos estudiados como explican [JMS10a] y [JMR10b].

La repetición y la precisión están muy ligados a los análisis científicos y a los entornos de pruebas. La herramienta se ha diseñado en general para poder minimizar la cantidad de tiempo que los investigadores del grupo CeNITEQ invierten en la comprensión y manejo de distintos programas o aplicaciones que les permiten realizar las diferentes configuraciones que requieren sus experimentos y el posterior tratamiento y análisis de los resultados.

A la vista de los requerimientos presentados en el capítulo anterior, y las recomendaciones del grupo sobre la compatibilidad, la modularidad, la facilidad de instalación y la necesidad de un interfaz amigable para el usuario, se han tomado importantes decisiones de diseño que se detallan a continuación.

El diseño e implementación de la herramienta tiene que permitir en concreto, su construcción en bloques o módulos y que se puedan utilizar cada uno de ellos de manera independiente, tiene que tener un manejo sencillo y permitir la modificación. Que sea portable entre diferentes sistemas Linux, no necesitar instalación o que la misma sea lo más sencilla posible. Que permita gestionar y utilizar las herramientas Tcpdump, Tc y Netem que son las usadas principalmente por los investigadores y que están en sistemas GNU/Linux.

2.1.1. Proyecto

Se va a definir ahora el concepto de “proyecto” y cómo se aglutina todo el trabajo o la herramienta bajo este concepto. La definición de proyecto es la estructura de organización creada para la herramienta y que engloba a todos los proyectos, capturas, escenarios y experimentos de una manera ordenada y coherente que facilitar el trabajo realizado. Se ha diseñado para que lo primero que pida la herramienta al ejecutarse sea un nombre de proyecto y que dentro del mismo se puedan realizar, almacenar y guardar un conjunto de experimentos con una o varias capturas en uno o varios escenarios que se vayan a realizar, todo esto se ha diseñado para que todo este organizado y ordenado para poder, ver, revisar, modificar o realizar las tareas propias de sus experimentos.

Este concepto de proyecto, permite diseñar la herramienta Mastercraft para una estructura de varios niveles, en el nivel superior está el fichero Mastercraft, que es el que pone en marcha la herramienta y a su mismo nivel se ha diseñado que aparezcan dos directorios, “lib” y “Proyectos”. En el primero, como vemos en la figura 2, se almacenan todos los script que utiliza la herramienta.



Figura 2 Directorio “lib” donde se guardan los script de Mastercraft

En el segundo directorio llamado Proyectos como se ve en la figura 3, es donde se guardan ordenadamente los proyectos creados por los investigadores. En un nivel inferior aparecen los directorios “Capturas”, “Procesar” y “Tabla”. Cada uno de estos directorios y subdirectorios se explica con más detalle en el punto 2.1.3 Estructura del Proyecto.



Figura 3 Directorio "Proyectos" con cada proyecto y sus subcarpetas generadas

2.1.2. Bash

Un punto importante que se tiene que decidir dentro del diseño e implementación, es sobre qué lenguaje de programación se va a implementar la herramienta Mastercraft, teniendo que basarse principalmente en los requisitos señalados al final del capítulo anterior, que para este punto serían:

Después de realizar un análisis de diferentes lenguajes de programación como son Java, C++, Ada, Perl o Python, se vio que cualquier implementación en ellos, requiere la instalación del programa, de librerías y un control de versiones dentro de las diferentes distribuciones que se utilicen y esas son entre otras las cosas que se quiere evitar. Por todo ello, se decidió programar la herramienta con Script de Shell. El Shell ofrece un lenguaje de programación interpretado que posibilita escribir, modificar y verificar programas de manera rápida, así como también la automatización de algunos procesos. Seguidamente se exponen los motivos de esta decisión de diseño.

El programa informático Bash (Bourne again shell), es un intérprete de órdenes y un lenguaje de programación de consola y se basa en la Shell de Unix, es el más fiable, común y más utilizado en casi todas las distribuciones de Linux, no requiere ningún tipo de instalación ni actualización de librerías. Es una interfaz consistente, muy potente y está por defecto en la mayoría de los sistemas Linux, además, posibilita la interacción entre el usuario y el sistema operativo. Incluye algunas sentencias básicas de programación que se emplean para la toma de decisiones, ejecución de ciclos y almacenamiento de valores en variables. [GNU15] [LSS15]

Bash permite con relativa facilidad, hacer ampliaciones y modificaciones de la herramienta y no tiene problemas con nuevas versiones del sistema operativo. Fedora y Debían son los sistemas Linux que suelen utilizar los investigadores del grupo. La mayoría de los scripts se pueden ejecutar por Bash sin ningún cambio, sin necesidad de instalación ni compilación, de fácil mantenimiento y es el más universal y el más conocido y usado por el grupo investigador entrevistado al ser entre otras cosas, potente, rápido y práctico.

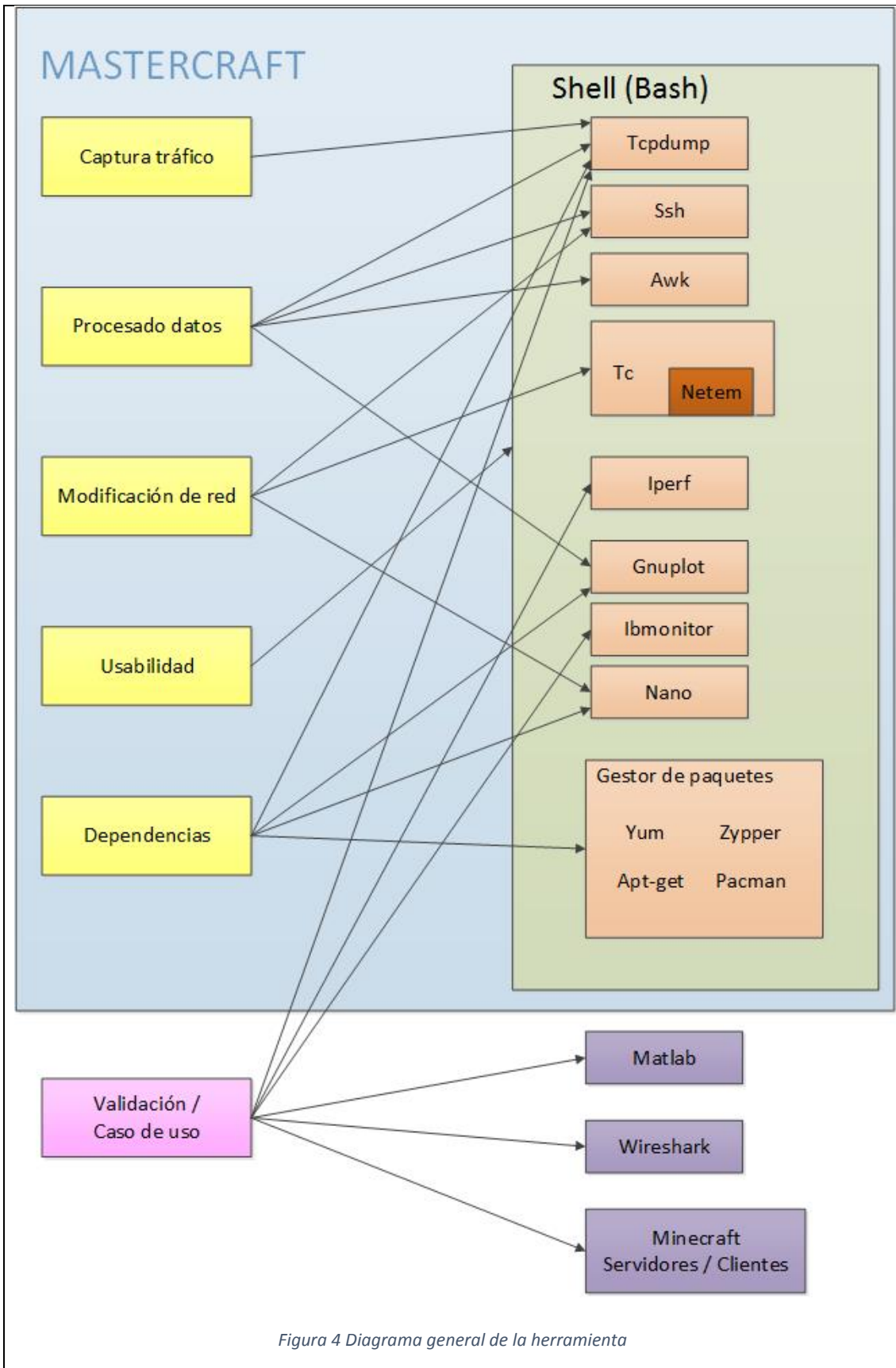
En conclusión, la utilización de Bash nos permitirá cumplir los requisitos exigidos por el grupo, como la creación de distintos proyectos, la facilidad de uso y de instalación, la modularidad e independencia de sus componentes, no tener que preocuparte de instalación, actualización o control de versiones en los distintos sistemas donde se instale. Además permite al investigador poder realizar los cambios que necesite o incluso poder gestionar desde la línea de comandos el sistema operativo.

Se presenta el esquema general de la herramienta que aparece en la figura 4 para poder mostrar los módulos de la herramienta Mastercraft que se ejecutan en la consola Bash y como interactúa con múltiples aplicaciones.

Como pequeño resumen para explicar la figura 4 indicar que Tcpdump se utiliza para realizar capturas y procesarlas, genera script en Bash para poder lanzarlo local y remotamente, comprobar su correcta ejecución y además con la ayuda del lenguaje de programación Awk, organizar y mostrar los resultados obtenidos, y con Gnuplot poder generar graficas de esos mismos resultados. Así mismo con Tc y Netem que cuentan con muchas opciones, y que tienen una complejidad elevada por su potencia en la modificación del tráfico de red y que se han integrado dentro de la herramienta, ya sea automatizando sus opciones más generales o permitiendo la ejecución de script con las opciones más avanzadas o específicas que un investigador pueda necesitar.

También indicar que como se ve en la figura 4 la validación y los casos de uso están fuera de la herramienta y que se verán con más profundidad en el capítulo siguiente, únicamente indicar el uso e instalación de clientes y servidores del juego Minecraft y el uso de Matlab a través de la generación de script propios que permiten mostrar unas gráficas de los resultados obtenidos en el procesamiento de los datos y así permitir la validación de la herramienta. De la misma manera que Wireshark con sus distintos filtros pueden verificar y validar los resultados generados por la herramienta. O los programas lbmonitor o lperf que monitorizan y generan tráfico artificial que se utiliza también para la validación de la herramienta.

Más adelante se explica de cada uno de estos programas y cómo se diseñó e implementó su integración y automatización dentro en la herramienta Mastercraft.



2.1.3. Estructura del Proyecto

Siguiendo con los criterios propuestos, para almacenar, organizar y ordenar todos los archivos que se necesiten, se ha optado por diseñar una estructura de directorios que permitirán tanto el acceso a los script modulares como una adecuada organización de todos los proyectos científicos realizados. Lo primero que tendrá que introducir el usuario al ejecutar la herramienta es el nombre con el que quiere que se guarden todas sus capturas, análisis y experimentos, este será el nombre de su proyecto, y bajo él se crea toda una estructura de directorios para mantenerlo ordenado y con un fácil acceso.

Como se muestra en la Figura 5 se ve el árbol de directorios de la herramienta, que se compone de un fichero script llamado "Mastercraft" (script principal) que será el que se ejecute, y es el que llama al resto de script. En la carpeta llamada "lib" es donde se encuentran el resto de ficheros script que realizan cada una de las partes o módulos de la herramienta, se ha diseñado para facilitar su modificación, así como la posibilidad de añadir nuevos módulos o funciones que permitan la mejora de la herramienta.

Al haber sido diseñada como una herramienta en el que cada uno de los script que tiene se pueda utilizar por separado, permite que por ejemplo se utilice sólo el módulo de capturar o el de procesar tráfico de red, o únicamente la realización de conexiones de red remotas o el módulo de la modificación de tráfico de la red. Esto posibilita que se puedan utilizar una o varias partes de la misma. Además es altamente ampliable y escalable, permitiendo de una manera sencilla, colocar sus script dentro de la carpeta de script "lib" y añadirlos como una opción más al menú que contiene el script Mastercraft.

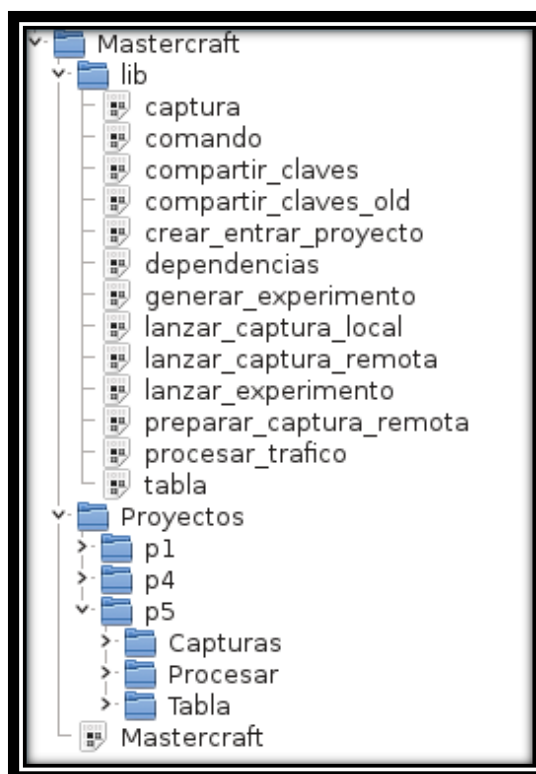


Figura 5 Estructura de la herramienta

Y por último una carpeta “Proyectos”, donde se generan y guardan todos los ficheros de cada uno de los proyectos de los investigadores, ordenándolos en sus respectivas carpetas. Cuando se crea un proyecto nuevo y se le da un nombre, como se ve en el ejemplo de la Figura 5, el diseño de la herramienta genera con dicho nombre una organización de carpetas y subcarpetas que facilitan las tareas de la gestión de datos. En una subcarpeta llamada “Capturas” se guardarán todas las capturas realizadas, en “Procesar” se guardan todos los análisis de las capturas y la información resultante, se procesa y muestra por pantalla y finalmente, en “Tabla” se guardan las tablas y script de los experimentos.

Además el diseño facilita la posibilidad de entrar en otro proyecto distinto del que estamos, para revisar o ejecutar cualquier opción que se necesite, y que al salir de dicho proyecto se quedará en el que se estaba anteriormente, todo ello sin tener que salir y luego volver a entrar en la herramienta. Todas estas opciones facilitan la conexión entre distintos proyectos, el aprendizaje y hacen mucho más sencillo y potente el manejo de la herramienta, como se pedía en los requisitos vistos al final del primer capítulo.

Dependencias

Se ha diseñado el módulo “dependencias” como se puede ver en la figura 4, que permite instalar automáticamente las dependencias que necesita la herramienta, en este momento son estos tres programas: Nano, Tcpcap y Gnuplot. Este diseño asegura que se pueda utilizar en los siguientes sistemas Linux: Redhat, Fedora, Debian, Ubuntu, Suse, OpenSuse, archlinux, chakra. O en cualquier sistema operativo Linux que tenga cualquiera de estos gestores de paquetes para la instalación de software: Yum, Apt-get, Zypper o Pacman. En conclusión, el diseño realizado, permite que la herramienta pueda correr en la mayoría sistema operativo Linux.

Comandos o programas externos

Para darle mayor potencia a la herramienta se diseñó de tal forma que permitiera “sin tener que salir de la misma” ejecutar programas u órdenes de línea de comandos, las mismas que te permitiría ejecutar desde la consola de Linux. La herramienta permite, listar directorios, cambiar permisos, crear usuarios, lanzar generadores de tráfico, etc.

Conexión remota

Otro de los requisitos que había que cumplir era facilitar la conexión automática con equipos remotos, la realización de capturas de tráfico de redes remotas para su posterior análisis, procesamiento y la posibilidad de modificar el tráfico de red en dichos equipos remotos. Para cumplir todos estos requisitos, se diseñó e implemento un módulo que permitiera realizar la conexión entre dichos equipos de una forma fácil, automática y segura. La herramienta se diseñó para que utilizara la orden Ssh dentro de su programación para crea conexiones seguras con los equipos remotos necesarios. Esta opción permite conectarse a cualquier equipo con sistema operativo Linux de manera segura a través del algoritmo RSA. Este es un algoritmo asimétrico que cifra en bloques, utiliza una clave pública, la cual es la que distribuye y otra clave privada, que es la que guarda en secreto el propietario, obteniendo de esta forma una conexión completamente segura.

Como resultado con este diseño se consigue eliminar la necesidad de introducir la contraseña cada vez que se realice una conexión remota. Este punto es muy importante para las capturas remotas automatizadas y para la modificación del tráfico de red en equipos remotos como se verá en el punto siguiente.

2.2. Captura de tráfico de red

Según los requerimientos presentados, la herramienta tiene que ser capaz de capturar, tanto en local como en remoto, el tráfico de red que se le pida, almacenarlo dentro de un proyecto y posteriormente tratarlo de manera ordenada. Para cumplir dichos objetivos, se analizaron diferentes opciones, como por ejemplo Tcpcmdump y Wireshark.

Se ha decidido utilizar Tcpcmdump porque como se puede ver en [TD15], es una potente utilidad para la captura de tráfico de red. Permite capturar y mostrar en tiempo real los paquetes transmitidos y recibidos en una interfaz de red. No tiene interfaz gráfica, sino únicamente línea de comandos, con lo que requiere de muy pocos recursos para su utilización y es idónea para una captura de paquetes de datos de forma desatendida. Funciona para la mayoría de sistemas operativos Unix: Linux, Solaris, BSD, Mac OS X, HP-UX y Aix entre otros y se requieren permisos de administrador para su utilización. Su última versión es la 4.7.4, a 22/04/2015. Existe una versión del mismo, llamado WinDump para los sistemas Microsoft Windows.

La conclusión final es que el grupo investigador quieren utilizar Tcpcmdump para sus experimentos por su familiaridad con el programa y su facilidad en la gestión de las capturas remotas así como por su potencia y velocidad ya que se ejecuta en línea de comandos y tiene una gran cantidad de parámetros además de la posibilidad de ejecución desatendida.

Wireshark resulta muy útil para la validación ya que permite de una manera sencilla la importación de ficheros capturados y al tener un entorno gráfico amigable da la posibilidad de sacar gráficas de resultados de manera bastante rápida y sencilla. Como se utiliza la primera herramienta para el diseño, se utilizará la segunda para su validación, como se ve en [WS15], y se verá en el capítulo siguiente.

En un primer momento se diseñó y programó la herramienta para que realizar únicamente capturas de red en el equipo local, teniendo que instalar Mastercraft en cada uno de los equipos en los que se quisiera realizar una captura, porque en los requisitos se pedía una fácil conexión a equipos remotos y la modificación de su tráfico de red, pero no la captura de tráfico remoto. Pero en versiones posteriores se vio la posibilidad de modificar el diseño para permitir capturas remotas multiplicando de esa manera su potencia y usabilidad, haciendo posible que sólo se tenga que instalar la herramienta en un equipo y que desde este, se controle la captura en equipos remotos y de manera desatendida y transparente se recupere la captura y se procese localmente.

Se diseñó la captura local para que fuera lo más sencilla posible para el investigador, que puede ver por pantalla todas las interfaces de red que posee el ordenador, tanto redes cableadas como inalámbricas y permite elegir fácilmente a través de cuál de ellas quiere realizar la captura de tráfico de red. El investigador puede dar el

nombre que quiera para la captura que se realiza y cuando acabe se guardará en la carpeta Capturas que generó la aplicación automáticamente cuando el investigador dio nombre al proyecto en el que trabaja.

Obviamente, la captura remota no podía ser tan sencilla como la captura local, pero se diseñó para que el investigador sólo tuviera que introducir los siguientes datos: nombre de usuario remoto, la IP o host remoto, y la interface de red del equipo remoto donde se quiera realizar la captura. Además se potenció la herramienta para permitir que se pudieran ordenar capturas en diferentes equipos remotos a la vez. La herramienta necesita que se le indique el tiempo total durante el que tiene que realizar la captura.

Como se puede ver en el ejemplo de la figura 3, dentro del proyecto “Manual” en la carpeta capturas nos aparecería una captura local que sería captura1 y una captura remota que sería “Captura2_10.3.12.98_eth0”. Como se puede ver, el nombre de captura remota tiene incluido la IP del equipo remoto y la interfaz donde se ha capturado, esta metodología implementada, asegura poder hacer capturas en el mismo equipo remoto en tarjetas de red distintas, por ejemplo Ethernet y wifi y que las guarde con distinto nombre en el equipo local.

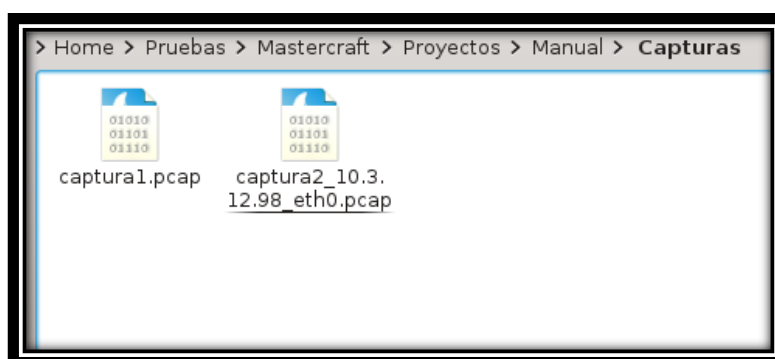


Figura 6 Ejemplo de capturas realizadas

Como conclusión, al haber diseñado la herramienta de esta manera, no se pierde tiempo en realizar las conexiones cada vez, ni tener que ejecutar comandos de Ssh y Tcpcdump con sus parámetros manualmente, ni en tener que dedicarse a ir a buscar en los equipos remotos las capturas realizadas y junto con las capturas locales ordenarlas y guardarlas en carpetas y todo ello manualmente, con la posibilidad de cometer cualquier error en cualquier punto y que tenga que repetir todo el proceso.

Una vez capturados los datos se puede pasar al procesamiento de los mismos y a la extracción de la información que el investigador necesite.

2.3. Procesado de datos

Según los requerimientos de los investigadores, la herramienta tiene que ser capaz de procesar las capturas filtrándolas por los distintos parámetros que se necesiten permitiendo para ello incluir expresiones complejas. Se utilizará como base el programa Tcpcdump igual que en el punto anterior, captura de tráfico de red.

Lo más usual será procesar una única captura aunque habrá veces que se tenga que procesar y sacar rápidamente resultados de múltiples capturas, para ello se ha diseñado e implementado la herramienta para que permita procesar una sola captura o todas las capturas. La primera opción, “una captura”, se ha diseñado para que procese y saque los resultados y estadísticas de una sola captura de nuestra carpeta “Capturas”, del proyecto que se tenga abierto en ese momento. En la segunda opción, “todas las capturas”, se procesaran todas las que encuentre en dicha carpeta. Por ejemplo si se hacen capturas remotas, se pueden procesar todas, una detrás de otra y poder analizar los resultados de una manera más rápida.

Para la generación de los script de procesado se ha decidido utilizar el programa *Awk*, que es en sí mismo un potente lenguaje de programación en línea de comandos. *Awk* ya viene incorporado dentro de Shell y está especialmente diseñado para trabajar con archivos estructurados y patrones de texto; permite modificar archivos, busca y transformar bases de datos, generar informes simples y otras muchas cosas. Lo que aporta principalmente a nuestra herramienta es que dicho lenguaje permite procesar secuencialmente archivos de texto como pueden ser las capturas del tráfico de red. En este caso los ficheros con las capturas de tráfico de red son procesados y modificados para poder extraer de ellos los datos necesarios y poder mostrarlos en otros ficheros que genera, así como en gráficas para su mejor comprensión. Su sintaxis es muy parecida al lenguaje de programación C, aunque sin la necesidad de tener que compilar nada. Funciona para cualquier Shell y se puede integrar sin ningún problema dentro de nuestros scripts de programa. [GRY16] [AWK16]. Con este único comando o herramienta o utilidad, podemos realizar todo el procesado de las capturas, sin necesidad de ningún otro comando.

Como podemos ver en la Figura 4 se ha diseñado la aplicación para que permita realizar el filtro que más les convenga, para eso se les proporcionan varias opciones. Filtrado por puerto, filtrado por IP y filtrado Manual.

```
Elige opción para el procesado
1) Procesar una sola captura
2) Procesar todas las capturas
3) Salir

Seleccionar número [1-3] 1
Qué captura quieres procesar?:
1 captura1.pcap
2 captura2_10.3.12.98_eth0.pcap

Digita el número de la captura: 1
Elige opción para filtrar las capturas
1) Filtrado por puerto
2) Filtrado por Ip
3) Filtrado Manual
4) Salir

Seleccionar número [1-4] █
```

Figura 7 Tipos de procesado, por puerto, por Ip y manual

En el filtrado manual, el más potente de los tres, se puede poner cualquier parámetro que necesite al *Tcpdump* que se ejecutará dentro de la herramienta y ésta se encarga

de mostrar y ordenar los resultados como en las opciones anteriores. El filtrado por puerto y por IP suelen ser las más comunes.

Se ha diseñado la herramienta para que permita no tener que repasar uno a uno cada uno de estos archivos generados, sino que los muestre por pantalla de consola en forma de resumen auto contenido de todo lo que se ha procesado. La herramienta va mostrando información de cada uno de los pasos que va realizando mientras procesa. Indicara cómo se van descubriendo todas las IP de la captura, cómo se van generando los ficheros de tráfico de red de subida y de bajada, cuál ha sido el tiempo total de la captura, el número total de paquetes capturados, así como una completa y estructurada tabla con toda la información que necesitan para analizar de un solo vistazo y así poder acceder a los ficheros específicos con mayor precisión y velocidad. La tabla que genera la herramienta y muestra por pantalla, contiene los siguientes campos:

- IP de captura con su puerto
- Inicio de sesión por IP y puerto
- Fin de sesión por IP y puerto
- Tiempo total de la sesión
- Total bytes de subida por IP y puerto
- Total bytes de bajada por IP y puerto
- Total de paquetes de subida por IP y puerto
- Total de paquetes de bajada por IP y puerto
- Ancho de banda de subida por IP y puerto
- Ancho de banda de bajada por IP y puerto

También informa la herramienta de cuándo está realizando la generación de los distintos gráficos de ancho de banda instantáneo de las IP tanto de subida como de bajada, utilizando para ello dentro del script el programa Gnuplot. [GN16] Gnuplot es un pequeño programa en línea de comandos que permite dibujar gráficos usando una tabla de coordenadas (en formato sólo texto) que se le pasa por cada una de las IP.

Así mismo el diseño realizado, permite potenciar aún más la herramienta, al poder pasarle al comando de procesamiento Tcpcmdump que se utiliza internamente, cualquier parámetro que se necesiten y que no sean puerto o IP. Como por ejemplo si queremos que procese sólo el tráfico TCP, UDP, ICMP, ARP, etc. Con el parámetro "src" y una IP filtraría el tráfico que tuviera su origen en dicha IP y con el parámetro "dst", que tuviera esa IP por destino. Con el parámetro "not" más un tipo de tráfico como UDP nos procesaría todo el tráfico menos el UDP. También en este caso, en el Anexo 2 se puede encontrar más información sobre los tipos de filtrados que se pueden utilizar.

Cuando se termina el procesado, se utiliza el diseño de estructura de directorios generado por la herramienta para guardar todos los datos generados dentro de la carpeta "Procesar" y dentro de ésta a una subcarpeta con el nombre de la captura donde se quedan definitivamente para que se pueda hacer un análisis más profundo de los mismos posteriormente, además de los datos, tablas y graficas que la herramienta generó para el estudio previo y que se mostraron por pantalla.

En definitiva, cumpliendo los requisitos exigidos, se realizan capturas de tráfico de red pudiendo elegir localmente el dispositivo de captura, además permite realizar capturas múltiples y remotas, almacenarlas dentro del proyecto para su posterior tratamiento. Se procesan capturas filtrándolas con distintos parámetros, incluyendo expresiones complejas. Y se han diseñado módulos especiales para que puedan mostrar resúmenes que faciliten la rápida comprensión de los experimentos realizados y se muestran ejemplos sencillos para facilitar su aprendizaje. La herramienta Mastercraft ayuda al investigador a realizar distintas operaciones y organizarlas dentro de su proyecto, como se ha visto hasta ahora, todas sus capturas, tanto locales como remotas en la carpeta “Capturas” y todos los análisis, gráficas y tablas que la herramienta ha generado en la carpeta “Procesar”, ahora es labor del investigador interpretar y casar las conclusiones de dichos datos, gráficas y tablas.

2.4. Modificación del tráfico de red

Siguiendo con los requerimientos, la herramienta debe de ser capaz de modificar el tráfico de red de una manera automática y remota, además de permitir incluir expresiones complejas en las ordenes que los investigadores ya utilizan, de forma manual, para dicha modificación, que son Tc (*Traffic Control*) y Netem (Network Emulator). Para cumplir dichos objetivos, se estudiaron y probaron dichas órdenes y se diseñó la herramienta para poder incluirlas de tal manera, que facilite su uso más genérico, automatizando el procedimiento de su ejecución y permitiendo así mismo la inclusión de cualquier expresión compleja que se necesite.

Para poder modelar la conexión de red de cada cliente que se necesite dentro de su experimento se utilizan únicamente estas dos órdenes, Tc y Netem. La orden Netem es una, se podría denominar suborden de Tc, esto quiere decir, que para utilizar Netem siempre es necesario un Tc anterior en la misma línea, no así al revés. Los investigadores utilizan Tc que es la única orden que permite realizar las tareas de modificación de red que necesitan y dicha orden sólo está disponible para la plataforma Linux. Tanto Tc como Netem vienen también por defecto en la mayoría de las distribuciones de Linux (Fedora, Debian, OpenSuse, Gentoo, Mandriva y Ubuntu).

El comando Tc gestiona el tráfico de red en una interfaz determinada [JMS10a], [JMS10b], [JMS11a], [JMS11b]. El control de tráfico consiste en diversas operaciones [ma03]. Limita el ancho de banda total disponible a un valor fijo conocido. Limita el ancho de banda de un usuario, servicio o cliente. Maximiza el *throughput* de tráfico TCP en un enlace asimétrico. Reserva ancho de banda para un usuario, servicio o cliente. Prioriza tráfico sensible a la latencia. Administra y distribuye el ancho de banda disponible. [MTC01].

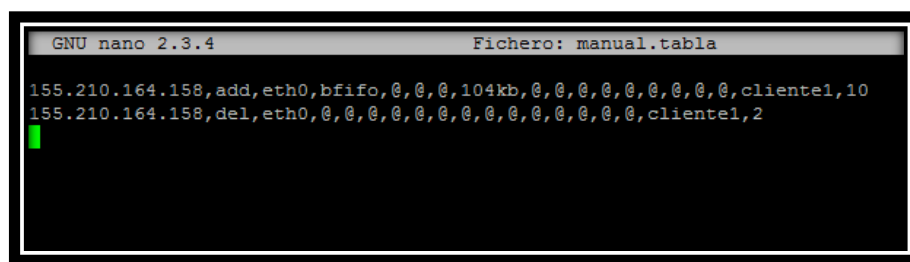
Además de la modificación del ancho de banda que se hace con Tc, se necesita modificar otros parámetros de la red y se necesita utilizar Netem. Netem [IF15] emulador de red dentro del kernel de Linux desde la versión 2.6.7. Netem es una extensión o un programa dentro del programa Tc, el cual está contenido en el paquete *iprouter2*. Netem nos permite introducir retardos, pérdidas y duplicación de paquetes en la transmisión de paquetes de cualquier interfaz [jms10b]. Netem se construye usando

la calidad de servicio existente (WOS) y las facilidades de diferenciación de servicios (diffserv) en el kernel de Linux [NT11].

2.4.1. Preparar experimento

Se ha propuesto en el diseño, la utilización del concepto del fichero tipo tabla. Este tiene que ser rellenado por el investigador y contendrá los parámetros que la herramienta utilizará para automatizar la modificación el tráfico de red que se necesita para llevar a cabo los experimentos. Las modificaciones en el tráfico de red se realizan con los programas Tc y Netem.

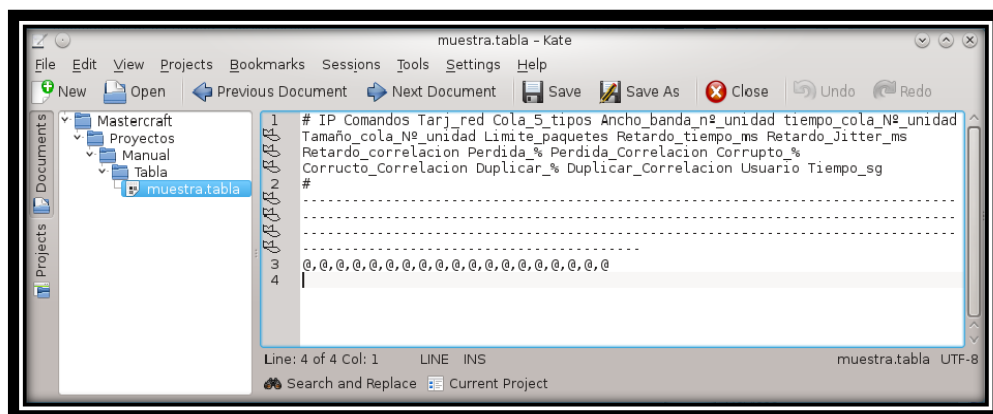
El diseño permite que en la opción preparar experimento, se pueda tanto crear como editar los ficheros con extensión “.tabla”.



```
GNU nano 2.3.4 Fichero: manual.tabla
155.210.164.158,add,eth0,bfifo,@,@,@,104kb,@,@,@,@,@,@,@,cliente1,10
155.210.164.158,del,eth0,@,@,@,@,@,@,@,@,@,@,@,@,@,cliente1,2
```

Figura 8 Ejemplo tabla sencilla creada por el investigador

Los ficheros tipo “.tabla” como el de la figura 5 contienen datos ordenados en columnas (separados por una coma) que le permiten a la herramienta generar los ficheros tipo “.exp” que después lanzará y realizará modificaciones del tráfico de red a equipos remotos. Como ejemplo y ayuda de uso, se ha diseñado que si en “Nombre de tabla o experimento:” se introduce: “muestra.tabla” se ve una cabecera indicativa de cada una de las columnas que se puede rellenar; como se ve en la figura 6.



```
muestra.tabla - Kate
1 # IP Comandos Tarj_red Cola_5_tipos Ancho_banda_nº_unidad tiempo_cola_Nº_unidad
Tamaño_cola_Nº_unidad Limite_paquetes Retardo_tiempo_ms Retardo_Jitter_ms
Retardo_correlacion Perdida_% Perdida_Correlacion Corrupto_%
Corrupto_Correlacion Duplicar_% Duplicar_Correlacion Usuario Tiempo_sg
#
.....
.....
.....
3 @,@,@,@,@,@,@,@,@,@,@,@,@,@,@,@
4
```

Figura 9 Muestra de ejemplo de Tabla editada con Kate

El diseño permite que si ya se ha generado algún experimento, dentro de esta opción del menú se puede editar y modificar, igual que se hace con las tablas, todas las veces que sea necesario. La herramienta abrirá el fichero que se le indique con un editor de texto dentro de la consola, este editor en línea de comandos es “Nano”, para que se pueda ver y hacer las modificaciones que se quieran sin tener que salir de la aplicación

y así poder ir ajustando la tabla para que posteriormente se genere el fichero del experimento que se ejecutara.

Después de un exhaustivo análisis de los programas Tc y Netem se ha diseñado la herramienta para que en los ficheros tipo “.tabla” que se generan estén compuestos por 19 columnas, (tantos como valores más habituales se utilizan en Tc y Netem) y a cada una se le pueda asignar el valor adecuado para configurar las herramientas Tc y Netem. A continuación en la tabla 1 se explica brevemente cada campo de la tabla, para una información mucho más detallada ir al Anexo 2:

Columna	Descripción	Columna	Descripción
1	Dirección IP	11	Correlación de retardo(<i>delay</i>)
2	Comandos Tc (add, change,...)	12	Perdidas % (<i>loss</i>)
3	Interfaz de red	13	Correlación perdidas (<i>loss</i>)
4	Colas: fifo, red, sfq, tbf	14	Corrupción % (<i>corruption</i>)
5	Ancho de banda	15	Correlación corrupción (<i>duplicate</i>)
6	Tiempo de cola	16	Duplicar % (<i>duplicate</i>)
7	Tamaño de cola	17	Correlación duplicar (<i>duplicate</i>)
8	Límite de paquetes	18	Nombre usuario
9	Tiempo de retardo (<i>delay</i>)	19	Tiempo
10	Jitter de retardo		

Tabla 1 Tabla explicativa de los campos de la tabla para Tc y Netem

Se ha diseñado la herramienta para que pueda automatizar la ejecución de estos comandos.

Una vez generadas las tablas necesarias, se puede pasar a la siguiente fase del experimento que sería convertir dichas tablas en script ejecutables.

2.4.2. Generar experimento

Se ha diseñado la herramienta para que en el apartado generar experimento, se conviertan automáticamente los ficheros tipo “.tabla”, en ficheros tipo “.exp” que son script de Shell preparados para ser ejecutados.

En la figura 7 se ve un ejemplo de fichero tipo tabla y en la figura 8 vemos el fichero tipo experimento que la herramienta genera automáticamente.

```
GNU nano 2.3.4 Fichero: manual.tabla
155.210.164.158,add,eth0,bfifo,0,0,0,104kb,0,0,0,0,0,0,0,cliente1,10
155.210.164.158,del,eth0,0,0,0,0,0,0,0,0,0,0,0,cliente1,2
```

Figura 10 Fichero "manual.tabla" de tipo tabla

```
GNU nano 2.3.4 Fichero: manual.exp
#!/bin/bash
sleep 10
ssh cliente1@155.210.164.158 sudo tc qdisc add dev eth0 root bfifo limit 104kb
sleep 2
ssh cliente1@155.210.164.158 sudo tc qdisc del dev eth0 root
```

Figura 11 Fichero "manual.exp" de tipo experimento ejecutable automáticamente

Tras la generación del fichero experimento, se puede pasar al punto siguiente que sería lanzarlo.

2.4.3. Lanzar experimento

Se ha pensado que dentro de este apartado, la herramienta tiene que permite antes de lanzar el experimento poder revisar todas y cada una de las conexiones a ip`s remotas que se necesitan para verificar que no nos pide contraseña y que el experimento va a realizarse automáticamente de principio a fin.

“Lanzar Experimento” lista experimentos generados y nos pide un nombre de experimento con su extensión para poder ejecutarlo.

De esta manera la herramienta cumple con los últimos requisitos exigidos por los investigadores, que eran facilitar la conexión automática a equipos remotos y permitir la modificación del tráfico de red de manera automática, mostrando ejemplos para facilitar su utilización.

Finalmente antes de entrar en la metodología de validación y en la realización de experimentos del tema siguiente, volver a recordar que en el Anexo 2 de este trabajo fin de master, se encuentra recopilado el Manual de Usuario/Programador donde se encuentra detallada cada una de las opciones del menú.

3. Validación y Casos de uso

Este capítulo se divide en validación y casos de uso, en la validación podemos ver cómo se fueron validando los diferentes módulos de la herramienta y qué otras utilidades o programas externos se utilizaron. En la definición de casos de uso se ven varios ejemplos de casos de uso para poder simular un experimento completo y cómo podrían llegar a resolverlo los investigadores y hasta dónde ayudaría la utilización de esta herramienta.

Cada una de las partes o módulos de los que se compone la herramienta Mastercraft, se fueron validando confirme se implementaban a través de otras aplicaciones como Wireshark, que verificaba si las capturas realizadas por la herramienta eran las mismas que las suyas, o las modificaciones que realizaba Mastercraft en el procesado de las capturas de tráfico de red daban los mismos resultados que los filtros introducidos en Wireshark para realizar de manera manual los mismos procesamientos.

Para comprobar el correcto funcionamiento de la herramienta, se han propuesto escenarios controlados de laboratorio con servicios reales de aplicaciones en tiempo real. Además, se utilizan generadores de tráfico que envían tráfico sintético con determinadas características que permiten la validación de funcionamiento y también diversos programas de captura, monitorización y análisis de tráfico. Las pruebas de validación que se utilizan en este trabajo son ejemplos similares a las pruebas que realizaría un investigador del grupo. No pretende ser una reproducción exacta o tan exhaustiva como las realizadas en las investigaciones, sino que su finalidad es únicamente validar la herramienta.

Se debe tener en cuenta que la herramienta fue diseñada para ser utilizada por los investigadores del grupo CeNITEQ, en sus distintos experimentos. En este capítulo se va a presentar ejemplos de casos de uso que muestre de forma completa los resultados que se pueden obtener con dicha herramienta. Sin embargo, tantos los posibles experimentos así como los escenarios necesarios para llevarlos a cabo pueden ser muy variados, dependiendo del tipo de estudio que se esté realizando. Los procedimientos que se van a seguir están basados en las metodologías del grupo CeNITEQ, estudiadas en el capítulo primero de este trabajo. En definitiva se ha seleccionado un caso concreto dentro de la actividad investigadora de esta área para mostrar la potencia de la herramienta.

3.1. Validación

3.1.1. Aspectos generales de validación

Al ejecutar la herramienta Mastercraft por primera vez o cuando un usuario introduce el nombre de un nuevo proyecto, se entra en un apartado de instalación de dependencias, estos son programas que necesita la herramienta Mastercraft que estén instalados para poder realizar sus operaciones. Estos programas son Tcpdump, Nano y Gnuplot. Para validar, que la herramienta realiza, cómo se le pide, la verificación e instalación automática de estas dependencias o programas, se realizaron diferentes instalaciones de sistemas operativos Linux en varios equipos físicos:

1. Ubuntu 14.04.2 LTS
2. OpenSuse 13.2 (Harlequin) (i586)
3. Centos 6.6
4. Fedora reléase 21

A estos sistemas operativos se les instaló una copia de la herramienta Mastercraft y se procedió a la instalación de las dependencias necesarias como hemos dicho anteriormente: Tcpcdump, Nano y Gnuplot. Seguidamente se comprobó su funcionamiento. Esta prueba validó esta parte de la herramienta y su perfecto funcionamiento en la mayoría de los Shell Bash de Linux.

Utilizando los sistemas instalados, también se procedió a validar la conectividad remota, para ello y utilizando la herramienta, se realizaron conexiones remotas (Ssh) entre los equipos utilizados.

La validación de las conexiones remotas realizadas por la herramienta posibilitó el paso siguiente a validar que es la captura de tráfico de red de forma remota. La herramienta realizó distintas capturas de tráfico de red remoto en cada uno de estos equipos y posteriormente como se había diseñado previamente entregó en el equipo local donde se estaba controlando la herramienta, todo el tráfico de las distintas capturas, validando de esta manera el apartado de captura.

3.1.2. Captura y Procesado de datos

Para comparar y validar los resultados que proporcionaba la captura y el procesamiento de la herramienta desarrollada, se utiliza Wireshark [WS15] que permite de manera sencilla y rápida, hacer una visualización de las capturas realizadas desde Mastercraft, y así ver rápidamente alguna de las características del tráfico de red.

Wireshark es software libre y se distribuye bajo Licencia Pública General de GNU (GPL), es un programa de análisis de protocolos de red. Se ejecuta en sistemas operativos, tales como Linux, Solaris, FreeBSD, Android, Mac os y Microsoft Windows. Conocido anteriormente con el nombre de Ethereal, este programa utiliza las mismas librerías de captura de paquetes que Tcpcdump y a diferencia de este tiene un entorno gráfico amigable aunque no permite la gestión de tráfico de forma desatendida, sí tiene en cambio unas potentes funciones de filtrado y análisis de tráfico. Permite la revisión de los datos mientras estos se están capturando o mediante la revisión de una captura ya almacenada. Requiere privilegios de administrador para el correcto funcionamiento de la gran cantidad de analizadores de protocolo que posee.

La herramienta al procesar la captura de datos, prepara los datos para que se puedan importar fácilmente para ser procesados por cualquier herramienta de software matemático, (por ejemplo para generar las gráficas que necesite sobre los datos capturados). Para validar esta funcionalidad se ha utilizado Matlab que es una herramienta de procesamiento de datos muy difundida. Entre sus usos se destacan: simular, modelar, crear prototipos, analizar datos y encontrar soluciones a sistemas

complejos. Se han creado algunos script de Matlab [MAT15] que procesan la información de las capturas y muestran las gráficas con los datos, ver anexo 3.

3.1.3. Modificación del tráfico de red

Para validar la modificación de tráfico de red, es necesario utilizar dos aplicaciones, una para poder monitorizar el ancho de banda en una determinada interfaz y comprobar que la red se ha modificado como se pedía, y otra para generar el tráfico artificial que necesitamos.

Para la monitorización de tráfico de red se utiliza lbmonitor [IB16] que es una aplicación interactiva de ejecutable en la consola. Esta es la que se ha utilizado durante las pruebas para comprobar aspectos cómo el ancho de banda utilizado y el total de datos transmitidos en todos los interfaces. También, se utiliza para comprobar el tráfico que transmite y recibe cada una de las interfaces de red de un equipo. Con esta herramienta se puede ver muy rápidamente cualquier variación de tráfico que se haya podido generar instantáneamente de manera precisa y efectiva.

Para generar el tráfico utilizaremos el generador de tráfico lperf [IP16a] [IP16b] que genera y envía tráfico artificial que nos permite validar el correcto funcionamiento de la herramienta. La arquitectura del mismo es del tipo cliente-servidor, por lo que se lanza en un equipo para que reciba y en el otro para que genere tráfico.

Se ha seleccionado porque es una herramienta muy difundida entre los programas de generación de tráfico, pues no solo lo genera, sino que también es capaz de monitorizarlo. Además realiza medidas punto a punto de ancho de banda, jitter, pérdidas y retardos de paquetes de tráfico de red. En cada una de las pruebas realizadas, evalúa entre otros datos, el ancho de banda y las pérdidas de paquetes.

Para todas las pruebas se utiliza la misma metodología basada en: definir un escenario de prueba que cumpla con los requisitos que se proponen, configurar el entorno físico necesario, realizar el experimento, comprobar si dicho experimento se ha realizado bien y si no es el caso repetirlo y por último presentar los resultados. Para realizar dicho experimento se siguen los 10 pasos del siguiente procedimiento:

1. Conectar equipos
2. Configurar equipos
3. Lanzar la herramienta
4. Lanzar servidor
5. Lanzar clientes
6. Empezar el experimento
7. Finalizar el experimento
8. Parar clientes
9. Parar servidores
10. Para la herramienta

3.2. Casos de uso

Para validar la capacidad de la herramienta Mastercraft para extraer información de las comunicaciones, por una parte, se eligió un juego real, actual, multitudinario y que se basa en la arquitectura cliente-servidor, y sobretodo es altamente configurable en el servidor, para crear las características idóneas dentro del juego y así poder capturar el comportamiento del mismo y poder estudiarlo.

Para esto, se va a realizar dos casos de uso: el primero de modelado de tráfico con dos experimentos que llamaremos 1A y 1B y el segundo un análisis de la gestión de red con otros dos experimentos dentro, que llamaremos 2A y 2B. Dichos casos de uso son ejemplos similares a las pruebas que un investigador del grupo realizaría para que sirvan como ejemplo.

Se realizan capturas remotas en los clientes y en el servidor y se muestran los resultados, como pueden ser: tiempo que permanecieron los jugadores conectados, número de bytes y paquetes enviados, el tiempo entre paquetes y el ancho de banda utilizado, también se puede observar tamaños de paquetes específicos según qué actividad concreta se realice dentro del juego por parte del jugador, etc. Es importante poder extraer toda la información que se pueda sobre el juego y poder ver el modelado del tráfico que genera. Esta información la utilizan los investigadores para conocer el comportamiento que tiene, en este caso, el juego.

Por otra parte, el objetivo final de las investigaciones que se quieren abordar están relacionados con análisis de la calidad (QoE es la calidad de la experiencia y QoS es la calidad de servicio) en las redes de comunicaciones. Por un lado, QoE como calidad subjetiva es difícil de medir, porque es la percepción que el usuario tiene de la calidad de un determinado servicio. Para hacer experimentos que midan la QoE se necesita la contribución de “investigadores sociólogos” que realicen diferentes test o estudios a los usuarios de los experimentos y vean cuáles son sus impresiones ante los distintos acontecimientos que se suceden en la red a lo largo del experimento. Por otro lado, QoS como calidad objetiva resulta más fácil de medir, principalmente ancho de banda, retardo, jitter (variación del retardo entre diferentes paquetes) y pérdida de paquetes. La QoS permite valorar el estado de una comunicación con la finalidad de repartir los recursos de red entre los diferentes servicios. Por lo tanto la responsabilidad de la herramienta es configurar la red con las condiciones esperadas, algo que se debe validar correctamente.

En resumen, además de las anteriores validaciones durante la programación de la herramienta, se van a realizar dos casos de uso: el primero es un análisis de modelado de tráfico y el segundo es un análisis de la gestión de red. Dentro de cada uno de estos casos de uso hay dos experimentos distintos. Y como se mencionó al principio de este capítulo, dichas pruebas o experimentos son ejemplos similares a las pruebas que un investigador del grupo realizaría para que sirvan como ejemplo.

Entrando en detalles, para realizar los siguientes casos de uso, se ha seleccionado una aplicación multimedia en tiempo real llamada Minecraft. Minecraft es un videojuego independiente de construcción, de tipo «mundo abierto» o sandbox, creado originalmente por el sueco Markus «Notch» Persson, y posteriormente

desarrollado por su empresa, Mojang AB. [MC15a]. Es un juego de exploración, recolección, creación y construcción libre a base de cubos (bloques), tiene distintos modos de juego como supervivencia, extremo, creativo, aventura o modos personalizados que crean los propios usuarios.

La elección de Minecraft se debe a que en 2015 se calcula que se habían vendido más de 20 millones de copias digitales en la versión Pc y Mac [MC15b], más de 30 millones en la versión Pocket y 12 millones de copias en Xbox 360, de PS3 no se conocen datos exactos, pero ha sido record de ventas desde su aparición y se calcula que el número de jugadores que tienen copias no legales de este juego en PC es mínimo tres veces superior al de legales. En total, uniendo todas las plataformas en las que se puede jugar a este juego, se superan los 60 millones de unidades vendidas. [GZ15] y [3DJ15].

Para las pruebas y los experimentos de este trabajo fin de master, se utiliza la versión tanto de cliente como de servidor desarrollada para PC, tanto en su versión para Windows como en su versión para Linux.

El juego está basado en la arquitectura cliente-servidor, como se ve en la figura 12 Para la creación de los mundos del juego se utiliza un algoritmo que proporciona una semilla numérica que es la utilizada para dicha generación. Si se utiliza esa semilla en distintos servidores, siempre se generará el mismo mundo, pero cada vez que se ejecute el algoritmo se generará una semilla distinta, por lo tanto, un mundo distinto.

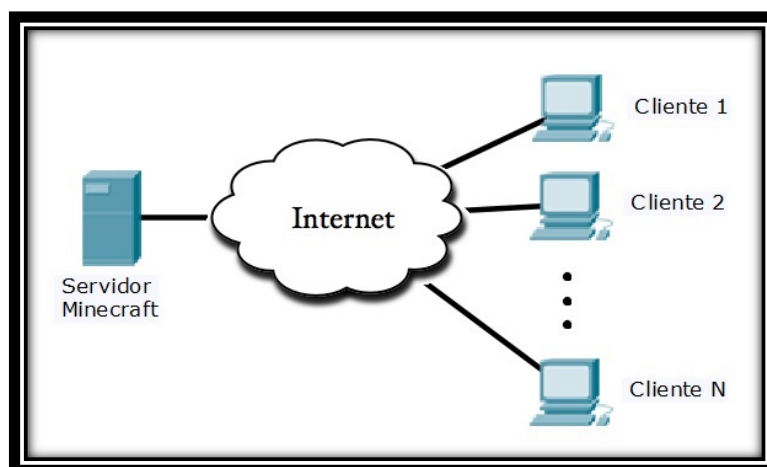


Figura 12 Juego Minecraft con un servidor y n clientes conectados a través de internet

Minecraft no utiliza el protocolo UDP para ninguna de sus comunicaciones como sí hacen otras aplicaciones del mismo tipo, sino únicamente TCP, los paquetes de TCP que los clientes envíen al servidor siempre tendrán una dirección IP fija, con un puerto concreto de servidor que previamente habremos configurado, y el servidor siempre enviará desde ese puerto. Los paquetes que envía a los clientes el servidor sí que van a distintos puertos. Este punto es importante porque para el procesamiento de los datos podremos filtrar todos los paquetes del juego con el binomio IP más puerto del servidor.

3.2.1. Caso de uso 1: Modelado de tráfico

Este caso de uso se divide en dos experimentos, el 1A y el 1B: en el primer experimento 1A, se utiliza la herramienta para realizar un análisis del tráfico generado en una partida multijugador, online. Este escenario ha sido escogido como ejemplo porque permite acotar el mapa de juego, indicar número de jugadores y tiempo máximo de juego y esto permite la repetitividad y automatización que se estaba buscando. En el segundo experimento 1B, se analiza el comportamiento del tráfico utilizando otra vez el juego Minecraft, pero esta vez no con una partida multijugador sino con capturas hechas a un único jugador en un mundo concreto realizando tareas específicas.

Tanto para uno como para otro, se han seleccionado unas configuraciones de servidor diferentes, que generarán mundos diferentes que harán que el juego se comporte de manera distinta. Este escenario ha sido escogido como ejemplo porque permite minimizar el tráfico que genera el mapa del juego y maximizar el tráfico que genera cada uno de los hitos que se ejecutan. Se captura con una cantidad idéntica de tiempo y así se pueden comparar el comportamiento del tráfico de red entre los diferentes hitos. Y los hitos que se van a capturar para poder analizarlos son 20 y se denominan de la siguiente manera:

1. Quieto. No mover el personaje.
2. Correr. Correr todo el tiempo, generando algo de mapa nuevo.
3. Caminar. Caminar todo el tiempo, generando más mapa nuevo.
4. Volar. Volar todo el tiempo generando mucho más mapa nuevo
5. Espada. Dar espadazos al aire.
6. Flecha. Tirar flechas.
7. Chat. Escribir frases en el chat del juego.
8. Cavar fila. Cava fila de tierra
9. Rellenar fila, Rellenar la fina que anteriormente se había cavado.
10. Comer.
11. Tnt. Explosiones cerca del personaje.
12. Morir ahogado. Tirarse a una piscina.
13. Morir lava. Tirarse a un rio de lava.
14. Morir altura. Tirarse desde un sitio alto.
15. Nadar. Nadar en piscina.
16. Cortar árbol. Cortar varios árboles.
17. Quemar zombie. En un cercado cuando se hace de día.
18. Matar zombie con flecha. En el mismo cercado.
19. Login. Hacer un login y sin hacer nada más posteriormente un logout.
20. Vida. Dejar que el personaje muera de hambre, sin moverlo.

Escenario de pruebas

Para el primer experimento, 1A, se configura el servidor con un modelo de juego conocido como “los juegos del hambre”, refiriéndose a la conocida película y que se recrea en este juego. Para ello, se instala y configura un servidor de Minecraft v. 1.6.4 Bukkit con el mapa Hunger Games (Juegos del hambre) como se ve en la figura 13.



Figura 13 Mundo Juegos del hambre Minecraft donde se ve mucha actividad entre los usuarios 1A

Este mapa que se genera, permite cargar siempre el mismo escenario, en las mismas coordenadas y con los mismos objetos siempre que le indiquemos un número concreto de semilla. Permite definir tamaño del mapa que se va a jugar, por ejemplo 150 bloques, 300 bloques, etc y no permite que los jugadores salgan de este tamaño de juego. El juego siempre comienza en el mismo sitio y nos permite definir el tiempo máximo de juego, siendo el escenario repetible infinidad de veces y sin reconfiguración alguna. El número máximo de jugadores es 16, aunque en este caso concreto utilizaremos 5, los clientes utilizar la versión cliente de Minecraft 1.8 y otras características como tiempo de invulnerabilidad o kit para los jugadores que no tienen relevancia aquí.

Para el segundo experimento, 1B, se instala y configura un servidor de Minecraft v. 1.8 Vanilla con el mapa genérico llamado mundo plano, figura 14, que se caracteriza por ser principalmente una zona o mundo infinito y completamente plano, sin montañas, ni cuevas, ni lagos, etc. Se compone de bloques de hierba, tierra y piedra base. Se configura también para que no aparezcan aldeas, monstruos, agua, lava ni ningún otro elemento que pueda generar tráfico entre el cliente y el servidor o viceversa. Esto posibilita que el tráfico generado por el mundo sea mínimo y que el tráfico generado cuando se realiza cualquier actividad (cualquier hito) sea mayor y pueda ser mejor acotado, definido y analizado.



Figura 14 Mundo Plano Minecraft donde el mundo casi no genera tráfico de red 1B

Como equipamiento hardware para la realización de las pruebas, se montan y configura los elementos que se ven en la figura 4, que son los siguientes: un servidor Minecraft Server: Core I5 con 4 Gb. ddr3 S.O. Windows 7, un equipo de control y análisis de red (en el que se ejecuta la herramienta Mastercraft) con procesador Core Dos Duo con 3 Gb ddr2 S. O. Fedora 21 a B4 bit y Kernel 3.19.3-200. Además, se utiliza un hub de 100 Mb Ethernet en vez de un switch para poder realizar las capturas más fácilmente y un router para gestionar el tráfico externo de la red. A dicho escenario, se conectan 5 Jugadores con clientes Minecraft 1.8 y equipo compatible con las características requeridas del juego y de la red y conectados en otra subred conectada a esta a varios *swich* y *router* de distancia. Todo el equipamiento hardware debe tener la suficiente capacidad para que no se produzca ni limitación ni alteración del tráfico de red.

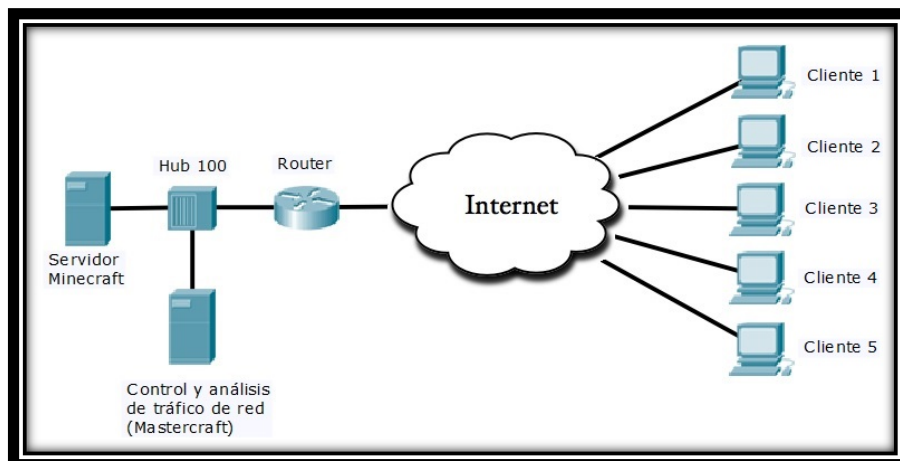


Figura 15 Escenario real Minecraft con un server y 5 clientes jugando y la herramienta Mastercraft

Resultados

Experimento 1A

Para el primer experimento de modelado de tráfico, 1A, se analiza el tráfico del juego de Minecraft online multijugador. Se realizó la captura y el procesamiento mediante la herramienta Mastercraft. Primeramente para el procesamiento, se filtró por la IP del servidor, que dio además de los resultados de captura, otro tráfico del servidor, como accesos de google, actualización del servidor y broadcast. Para eliminar estos resultados se procesó o filtro por un puerto seleccionando el 5050 que es el puerto por el que envía y recibe información el servidor de Minecraft. Aquí surgió el problema de al tener en la misma subred dos servidores configurados de manera parecida y que tenían como puerto el 5050, con lo que aparecían dos servidores aunque uno de ellos no participaba en el experimento ni generaba nada más que tráfico de espera de login. Finalmente y como permite la herramienta, se procesó por IP y puerto y ahora si mostró los resultados deseados como se puede ver en la figura 16. En dicha figura se puede observar, las IP con sus puertos, el inicio y fin del tiempo de captura, el tiempo que permaneció cada uno de los clientes conectados, los bytes y paquetes que enviaron y recibieron y el ancho de banda instantáneo tanto de subida y de bajada que generan.

Nº	Ip_y_Puerto	Inicio_Sesion	Fin_sesion	Sesion	Bytes_Up	Bytes_Down	Packet_up	Packet_down	AB_up	AB_down
01	155.210.140.139.28090	1430312964268606	1430314500561856	1536293250	1814161	9056105	32625	34873	0.514965	8.84329
02	155.210.140.139.29720	1430314512408885	1430314515400608	2991723	80	0	3	0	106.667	0
03	155.210.140.139.49013	1430312925338523	1430312925377774	39251	264	247	6	5	53.4568	64.2743
04	155.210.140.139.49016	1430312927397466	1430312927425622	28156	2417	327	60	7	162.853	46.4478
05	155.210.140.140.50437	1430312945476130	1430312945575156	99026	2784	407	69	9	141.426	35.8342
06	155.210.140.140.50438	1430312963047889	1430313730790686	767742797	1092286	8110399	17845	19698	2.15829	14.8777
07	155.210.140.140.50489	1430313852250116	1430313852284257	34141	2784	437	69	10	144.119	32.2927
08	155.210.140.160.53362	1430312963146355	1430314490486543	1527340188	1969461	10463879	36411	38102	2.15269	9.15639
09	155.210.140.160.55913	1430314492998127	1430314493023880	25753	264	237	6	5	53.7987	64.2765
10	155.210.140.8.60569	1430314563710358	1430314563710358	0	40	0	2	0	160	0
11	155.210.140.8.61067	1430312962815451	1430314487607451	1524792000	1625614	7972478	32645	33898	1.20018	5.83147
12	155.210.164.42.54306	1430312600644473	1430312600682673	38200	2674	2207	65	54	116.993	227.807
13	155.210.164.42.54325	1430312963133102	1430314486959863	1523826761	1590313	6274553	31638	32823	1.2354	5.13715
14	155.210.164.42.54371	1430314561491629	1430314564490826	2999197	80	0	3	0	106.667	0
15	155.210.164.46.5050	1430312600644473	1430314564490826	1963846353	41881276	8103222	159474	151434	28.7258	3.89834

Figura 16 Resultados presentados por pantalla para el primer análisis del investigador

Se programó un script en Matlab que utiliza los datos capturados para poder generar unas gráficas más elaboradas, que el investigador requiere para extraer sus resultados En la Figura 17 se representa el ancho de banda instantáneo tanto de subida (Uplink) como de bajada (Downlink) para los 5 clientes (cada uno de un color diferente). En el eje Y se ve el ancho de banda en *Mbps* y en el eje X el tiempo en microsegundos. Se aprecian picos muy altos debido a que se producen ráfagas de paquetes muy pequeños, es decir, se generan paquetes muy seguidos (con muy poco tiempo entre paquetes) por lo que el ancho de banda instantáneo aumenta considerablemente.

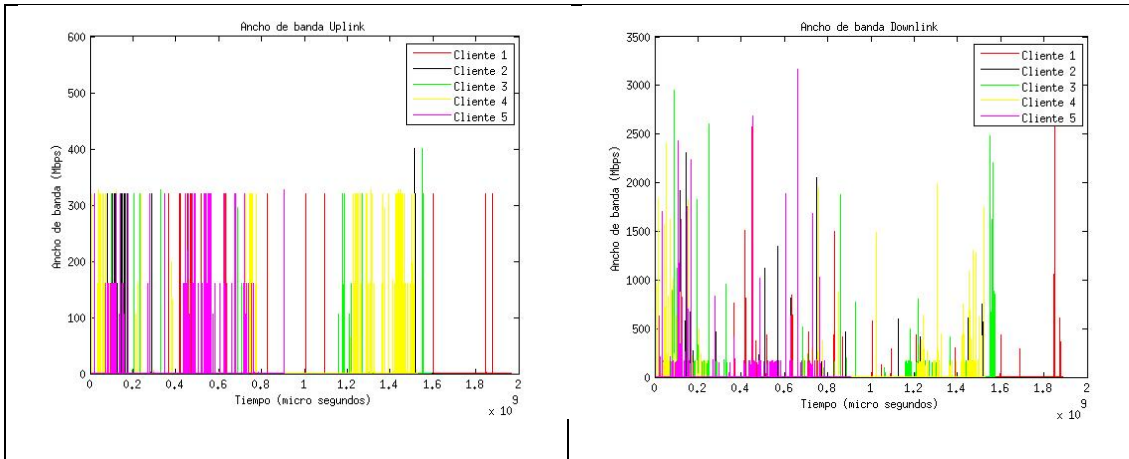


Figura 17 Ancho banda de los 5 clientes Up y Down

Otra forma de representar los mismos resultados y validar los datos que se obtienen de la herramienta Mastercraft en conjunto con el procesamiento que se hace en Matlab, es la gráfica que se muestra en la Figura 18. En dicha figura se representa en el eje Y el ancho de banda instantáneo de subida (Uplink) de cada cliente en *kbps* y en el eje X el tiempo en *segundos*, en este caso los datos han sido tomados directamente desde la captura en Wireshark y ambas figuras, 17 y 18 son similares. Por tanto, se valida y comprueba que la herramienta Mastercraft captura y procesa los datos correctamente.

En la figura 18 se puede observar el ancho de banda instantáneo consumido por los 5 clientes, presentando picos de hasta 35 Kbps. La media de ancho de banda para cada uno de los clientes es de aproximadamente 11 Kbps, los 5 clientes enviarían al servidor alrededor de 55 kbps y el servidor les envía 186 kbps, porque además del tráfico normal del juego, el servidor envía el mundo generado a cada uno de los clientes.

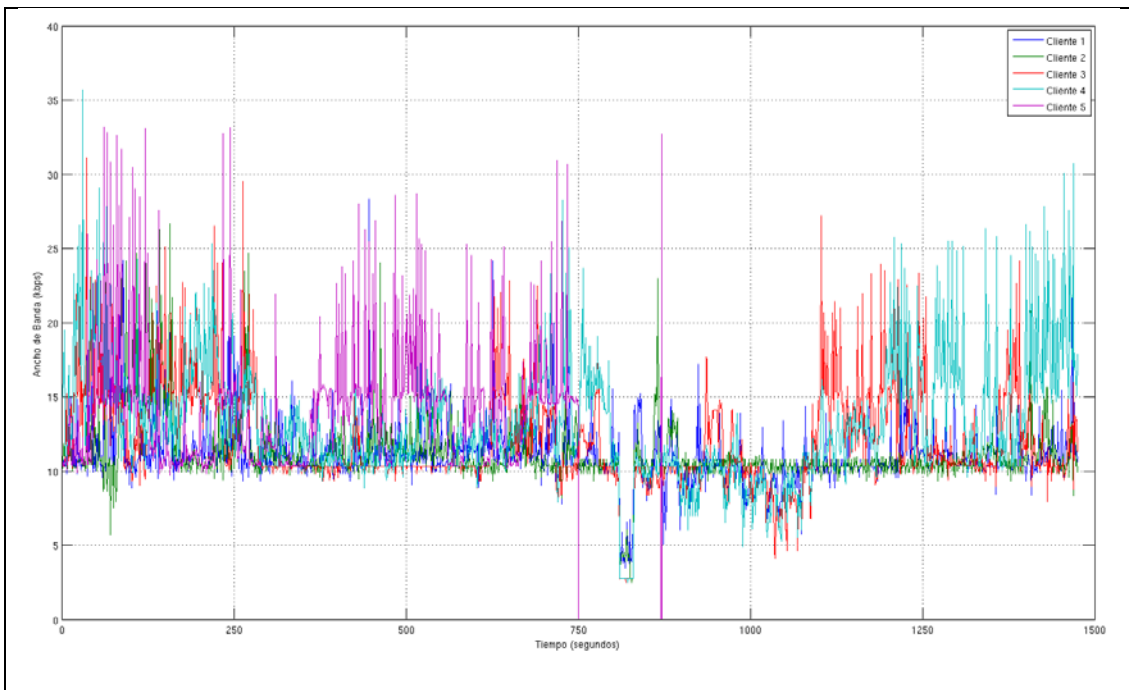


Figura 18 Ancho banda de subida para 5 clientes Wireshark

En la figura 19 se representa los paquetes por segundo que envían tanto el servidor como los clientes. Hay situaciones en las que es necesario obtener los resultados en paquetes por segundo dadas las limitaciones tecnológicas. En el eje X se representa el tiempo en segundos y en el eje Y el número de paquetes por segundo. Se puede observar que los clientes envían muy pocos paquetes, lo que se relaciona con el poco ancho de banda consumido dado que los tamaños de paquetes son similares. Sin embargo, el servidor envía hasta 3 veces más paquetes por segundo en media, que la suma de todos los clientes, concurda también con la diferencia de anchos de banda entre servidor y clientes.

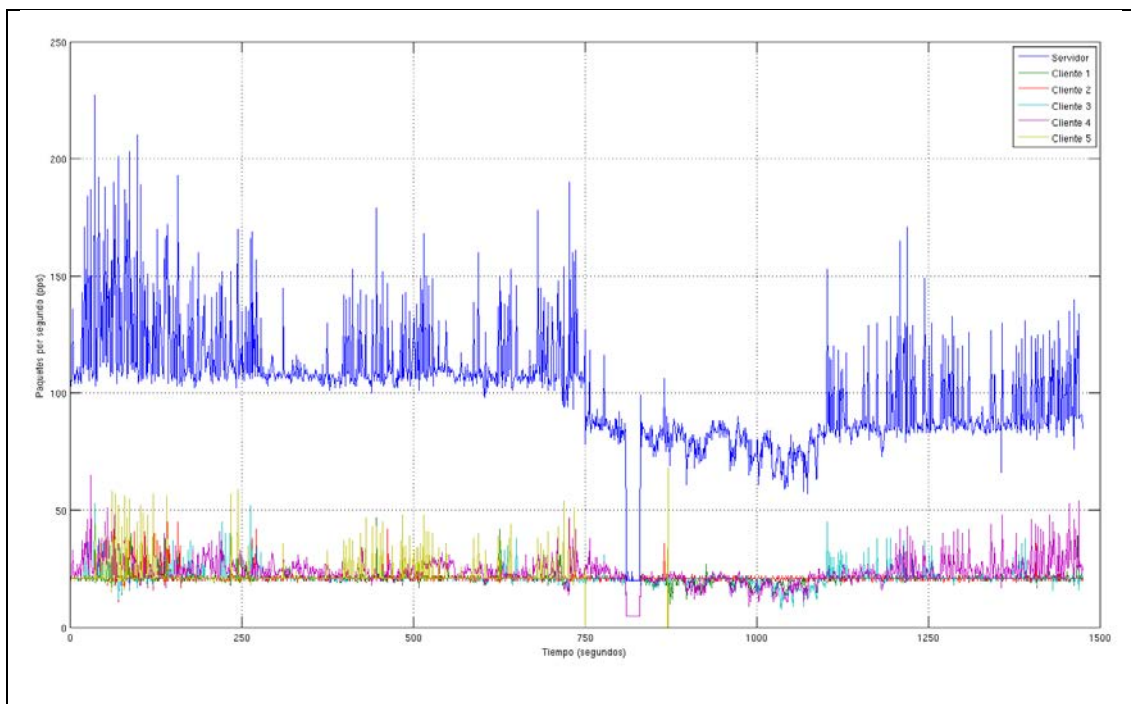


Figura 19 Paquetes por segundo de subida de servidor y clientes Wireshark

La justificación que tiene este experimento es la importancia del estudio de caracterizar las funciones para identificar en que franjas horarias se realizar las distintas acciones.

Experimento 1B

Para el segundo experimento 1B, dentro del caso de uso de modelado de tráfico, se analiza el tráfico del juego de Minecraft. Se realizaron 20 capturas de distintos comportamientos en el juego, entre servidor y cliente, durante un tiempo concreto (60 segundos cada una) y esto mostrará la diferencia en el tráfico de red dependiendo de la actividad realizada, por ejemplo permanecer quieto sin hacer ninguna actividad, caminar, correr, volar, lanzar flechas, etc.

En esta prueba o experimento se muestran algunas características del tráfico, centrándose en el tamaño de los paquetes transmitidos. En la tabla 2 se observa el tráfico de red generado cuando el cliente envía al servidor paquetes de un tamaño determinado en cada una de las acciones siguientes, se ha dividido la tabla en tres partes para su mejor comprensión.

Tamaño paquete	Acciones					
	Quieto	Correr	Caminar	Volar	Espadazos	Flechas
43					186	
44	1149	1			1124	1028
46		2				
48	30	29	29	30	30	30
52	5				26	134
53						28
61						27
68	58	1203	1183	1213	58	54
76						4

Tabla 2 Distribución de tamaño de paquetes que envía el cliente al servidor en función de los hitos

En la tabla 3 se muestra el número de paquetes que envía el servidor al cliente. En el Anexo 3 se encuentran las tablas originales tanto del tráfico de cliente como de servidor.

Tamaño paquete	Acciones					
	Quieto	Correr	Caminar	Volar	Espadazos	Flechas
40	1240	1130	1137	1068	1228	1139
45						5
47						27
48	30	29	29	27	30	352
52		4				
55						3
56						8
59	60	58	57	54	60	60
62		2		2	2	2
64						2
66						8
68		28	24	51		
77 a 1000		164	133	207		87

Tabla 3 Distribución de tamaño de paquetes que envía el servidor al cliente en función de los hitos

En la figura 20 es un ejemplo de 6 de los 20 hitos que se han capturado, los que aparecen en la figura 6 y 7, que son quieto, correr, caminar, volar, espadazos y flechas. En el anexo 3, están todas las tablas y las gráficas de todos los hitos, estos son sólo un ejemplo de lo que se podría analizar. Por ejemplo se ve que los hitos que tienen movimiento, como son caminar, correr y volar, tienen el mismo tipo de paquetes, con la diferencia de que contra más rápido se va, los paquetes se multiplican, en los mismos 60 segundos hay más cantidad de paquetes contra más rápido vas. Los envíos de cliente a servidor en los tres casos, casi todos los paquetes son de tamaño 68 y manda 1 paquete de tamaño 48 cada 2 segundos. En los envíos de servidor al cliente muchos paquetes son de tamaño 40 y cada 1 segundo, manda un paquete de tamaño 59. (Esto último pasa con todos los envíos del servidor independientemente del hito realizado).

Sin embargo en otros hitos que no tienen nada o casi nada de movimiento, como son, quieto, dar espadazos al aire, flechazos al aire, escribir en chat o quitar vida, tienen el mismo tipo de paquetes con la misma cantidad. El envío de paquetes del cliente al servidor es igual para todos o muy similar, casi todos los paquetes enviados son de

tamaño 44 y también envían 1 paquete de tamaño 48 cada dos segundos. Quieto y quitar vida, son muy parecidos en su comportamiento, envíos de cliente de paquetes de tamaño 40 iguales, de 50 de 70. Y de tamaño 60 en el caso del servidor.

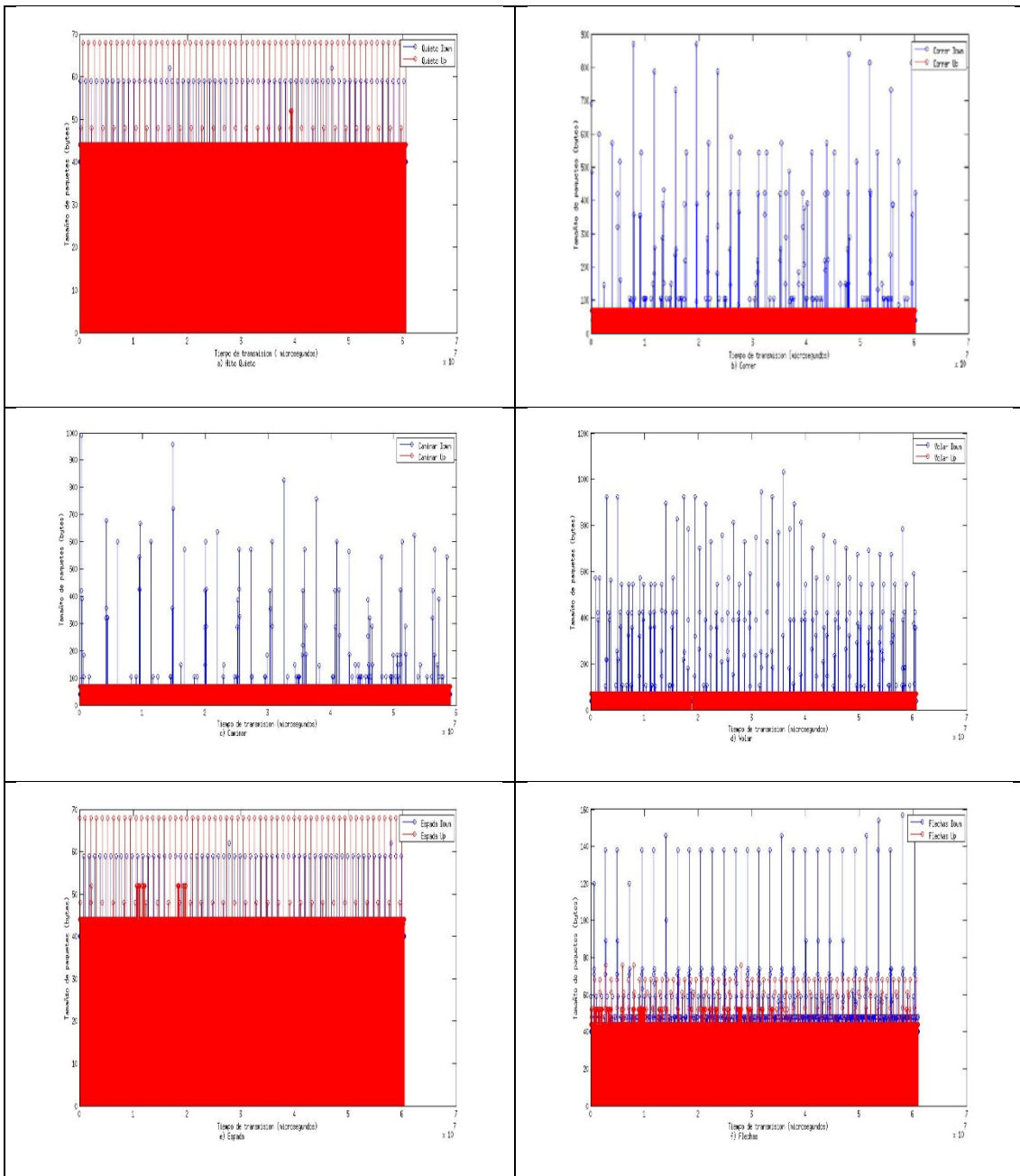


Figura 20 Graficas de los hitos, quieto, correr, caminar, volar, espadas y flechas

En la figura 21 se muestra como ejemplo el hito login, y se ve perfectamente el comportamiento del servidor, de color azul, con muchos paquetes de 400 bytes al comienzo, que es cuando el cliente hace login en el servidor, y le envía el mapa del juego, luego el cliente envía paquetes de tamaño 44 bytes cada cierto tiempo para indicar que sigue vivo en el juego y se ve el logout del servidor con un solo paquete de 100 bytes al final, que es cuando se desconecta el cliente.

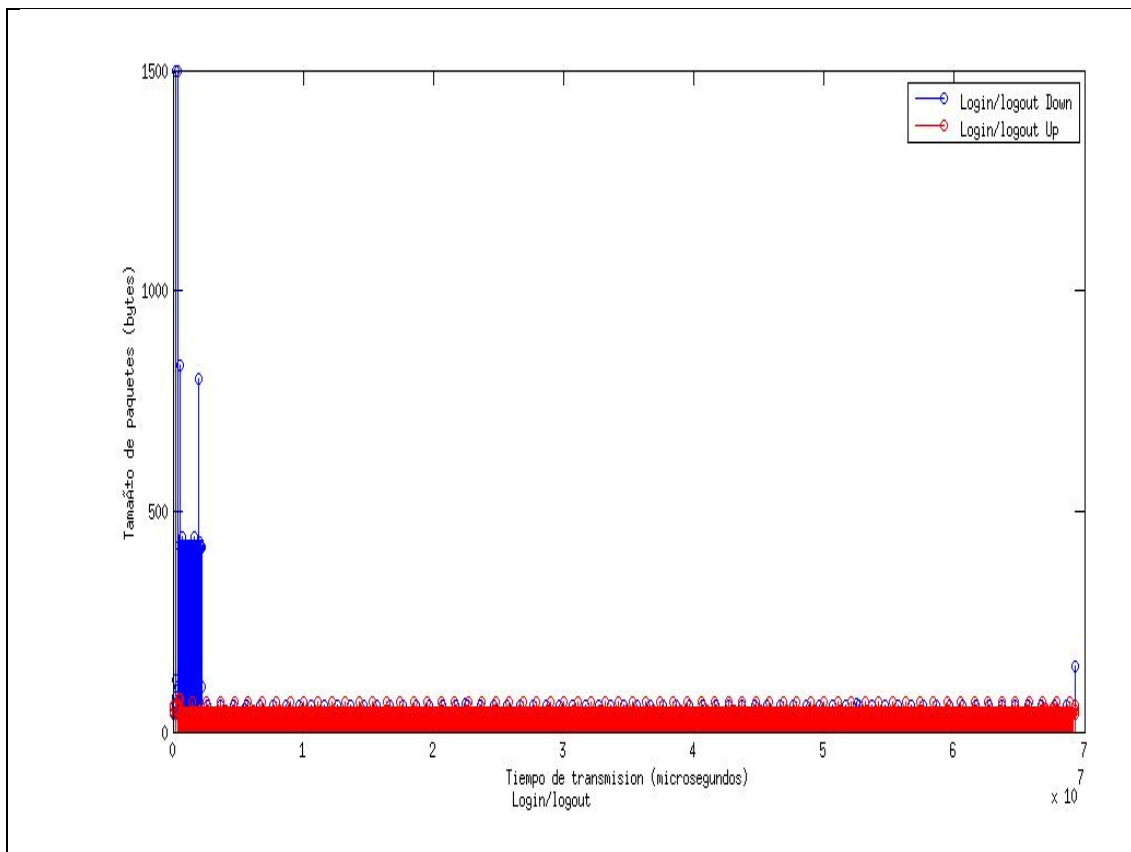


Figura 21 Gráfica del hito Login con tamaño de paquetes en bytes en función del tiempo en segundos

3.2.2. Caso de uso 2: Gestión de red

Para este segundo caso de uso, se realizan también 2 experimentos llamados 2A y 2B. En el primer experimento, 2A, se utiliza la herramienta para realizar una limitación de un tráfico simulado entre el servidor y un cliente similar al que se tendría en el juego. En el segundo experimento, 2B, se limitará un tráfico simulado entre el servidor y los 5 clientes del juego multijugador, online Minecraft. El objetivo de las pruebas es realizar el validado primero en un escenario más simple para luego complicarlo añadiendo nuevos elementos al escenario.

Con estos experimentos se debe comprobar la variación del ancho de banda que se gestiona en la interface de red por medio de la herramienta desarrollada. En este caso de uso se ha generado tráfico en la red para posteriormente limitarlo con la herramienta y ver que lo realiza como se le especifica. Por una parte, se va a limitar el

ancho de banda del enlace de salida del servidor y por otra se va a limitar el ancho de banda que reciben cada uno de los clientes con diferentes valores. De esta forma lo que se hace es emular el acceso de subida que tiene el servidor y los accesos de bajada que tienen los clientes.

Escenario de pruebas

Para el primer experimento, 2A, se prepara un escenario similar al que se ve en la figura 22, aprovechando así el escenario que ya hemos utilizado, donde se tiene a un servidor generando tráfico que recibe un cliente y mediante la herramienta hacemos que el tráfico de red que recibe el cliente se vea limitado de tal forma que en vez de recibir todo el tráfico que envía el servidor, reciba menos, indicándole cuánto menos.

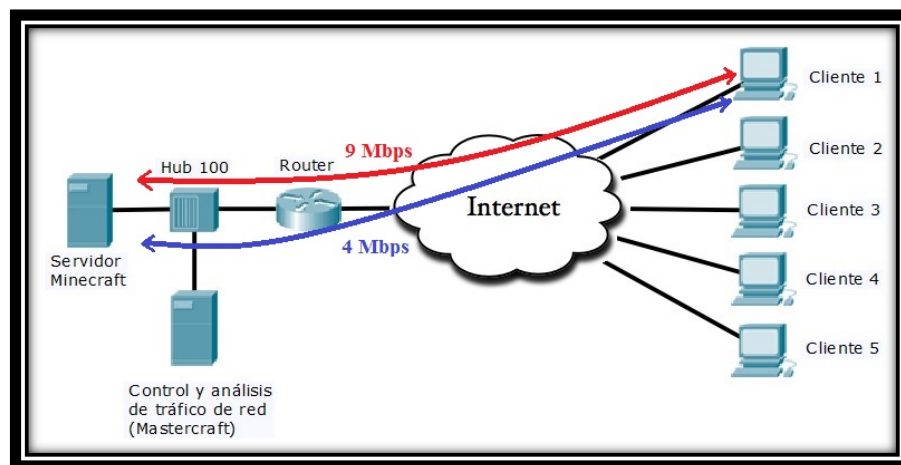


Figura 22 Escenario real donde se ve el tráfico generado a 9 Mbps y su cambio a 4 Mbps

En ambos extremos se lanza además, Ibmmonitor que es el monitor de tráfico de red que se utiliza para ver si el tráfico se genera y modifica de una manera instantánea y muestra la interface de red utilizada, los datos enviados, los recibidos y el total. Con esto se ve la interface que tiene el tráfico, cuanto tráfico envía, cuanto recibe y el total. Además lanzaremos (`Watch -n1 'sudo tc qdisc show'`), para ver cómo verdaderamente se modifica la interface de red que recibe el tráfico.

Para el segundo experimento, 2B, se prepara un escenario similar al que se ve en la figura 23, aprovechando también el escenario ya existente, donde se tiene a un servidor generando tráfico que reciben los 5 cliente y mediante nuestra herramienta se hace que cada cliente limite el tráfico tal y como se haya definido en nuestra herramienta.

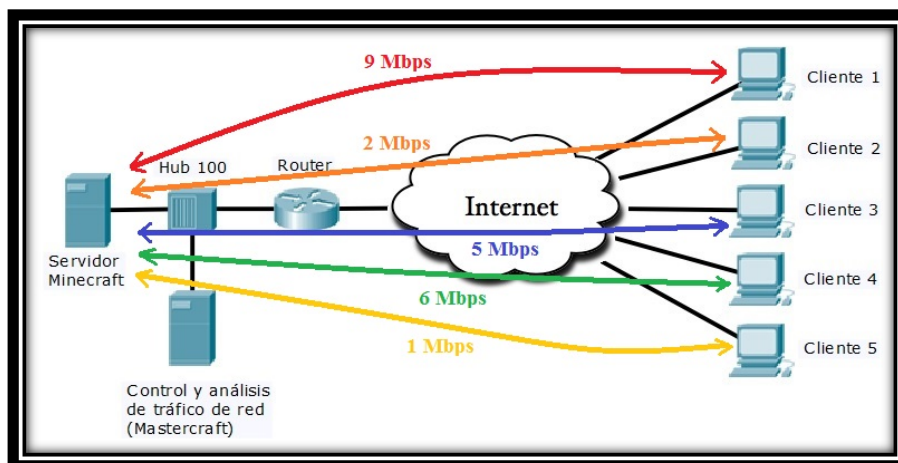


Figura 23 Escenario real donde se ve el cambio de tráfico en cada una de las comunicaciones

Resultados

Experimento 2A

Una manera muy sencilla de verificar que la herramienta funciona como se le pide, es utilizando Iperf para verificar que el tráfico que sale de la interfaz se vea limitado tal y como se ha configurado. Esto se consigue enviando con Iperf un ancho de banda mayor que el que se ha configurado como limitado en la interfaz. En concreto, se va a configurar la interfaz mediante la herramienta para que deje pasar 4 megas y con el generador de tráfico se va a enviar 9 megas.

Primero, lanzamos la captura de red desde nuestra herramienta Mastercraft, en un equipo distinto al emisor y receptor del tráfico generado. Seguidamente se lanza el cliente Iperf que es el que genera el tráfico, de la misma manera, que un servidor de Minecraft enviaría tráfico del juego a uno de sus clientes, con la siguiente orden (Iperf -c 155.210.157.65 -u -b 9M -t 50). Ahora, se lanza el servidor Iperf que es el que recibe el tráfico, igual que haría nuestro cliente del juego Minecraft, con la siguiente orden (Iperf -s -u -i1-f MB).

Con la herramienta se quiere modificar el ancho de banda y se ha creado la siguiente tabla.

```
IP_remota,add,eth0,tbf burst 5kb limit
150000,0.5mbps,@,@,@,@,@,@,@,@,@,@,@,@,@,proyecto,5

IP_remota,del,eth0,@,@,@,@,@,@,@,@,@,@,@,@,@,proyecto,20
```

Una vez creada la tabla con la herramienta ya puede generar Mastercraft automáticamente un script "ancho banda1.sh" que es el que se ejecutara para realizar las modificaciones de la red.

Con la tabla de Mastercraft generamos para el experimento, el script anchobanda1.sh aunque el investigador puede crear el script directamente:

```
#!/bin/bash
sleep 5
ssh proyecto@IP_remota sudo tc qdisc add dev eth0 root tbf burst
5kb limit 150000 rate 0.5mbps
sleep 20
ssh proyecto@ IP_remota sudo tc qdisc del dev eth0 root
```

Como se puede ver en el script se ha utilizado TBF el cual es útil cuando se desea que todo el tráfico que sale por una determinada interfaz tenga unas características determinadas en cuanto a ancho de banda, tamaño de cubeta y latencia. El algoritmo TFB es un limitador de tasa basado en pasar paquetes a la red con un token. Estos tokens se generan con una tasa constante, tasa a la que se quiere limitar la salida del flujo. Por tanto su aplicación es limitar la tasa máxima de salida de los flujos sin garantizar ningún ancho de banda mínimo o préstamo entre clases. [EM05]

Esto serviría para ver cómo cambia la calidad de red entre los equipos cliente y un equipo servidor, modificándole el tamaño del ancho de banda y viendo objetivamente y subjetivamente cómo afecta esto al usuario. Pero hay que tener en cuenta se cambiaría el ancho de banda a la salida de red del equipo servidor y afectaría a todos los clientes por igual. Si se quiere modificar el tráfico que reciben los clientes con parámetros de red distintos, (para emular velocidades de bajada distintas en cada acceso de cliente), por ejemplo a un cliente limitarle el tráfico a 1 Mb, a otro limitarlo a 4Mb y a otro a 7Mb por ejemplo, este método no serviría.

Como se ve en la figura 24 se realiza una captura de 60 segundos y se genera un tráfico de 9 Mb que permanece durante 20 segundos hasta que la herramienta a través del script que tiene lo reduce a 4 Mb durante 10 segundos para luego volver a dejarlo en 9 Mb. La captura realizada por nuestra herramienta a la que le damos el nombre "bw14_155.210.157.66_eth0.pcap", la pasamos por Wireshark para que nos muestre una gráfica del comportamiento del ancho de banda durante el experimento y validar que la herramienta ha realizado lo que se le ha pedido.

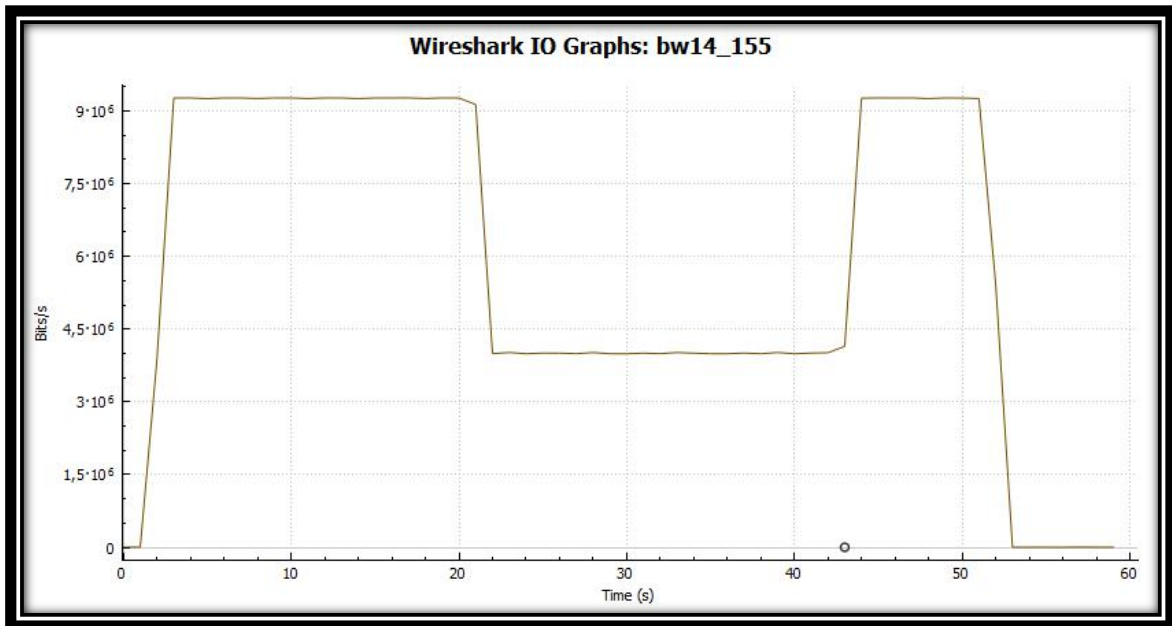


Figura 24 Grafica con la modificación del ancho de banda realizado con la herramienta Mastercraft

Experimento 2A

Para el segundo experimento, 2B, de este segundo caso de uso de gestión de red, lo que se requiere es gestionar a cada usuario de forma independiente y así poder hacer los cambios en el ancho de banda de red no en la tarjeta de red de salida de dicho tráfico, sino en el tráfico para cada uno de los clientes, se recomienda crear una disciplina de colas o qdisk con clases, cada clase a su vez se puede dividir en otras clases o subclases y a estas se les puede redirigir paquetes de tráfico de red a través de filtros. HTB es necesario cuando se quiere clasificar el tráfico que sale por una determinada interfaz en varias clases, cada una de ellas con unas características diferentes de ancho de banda. Si alguna clase no utiliza todo el ancho de banda que se le ha definido, dependiendo de la configuración, se podría utilizar dicho ancho de banda para alguna otra clase que lo necesite.

Con la herramienta, siguiendo los pasos de la prueba anterior y con el script que se muestra a continuación podemos modificar el tráfico de red que reciben los 5 clientes del juego, dándole por ejemplo 9 Mbps cliente 1, 2 Mbps cliente 2, 5 Mbps cliente 3, 6 Mbps cliente 4 y por ultimo 1 Mbps al último cliente. Este es solamente un ejemplo de cómo se podría modificar el tráfico con un script que genere la herramienta y que el “investigador sociólogo” responsable de la prueba modifique para que se ajuste a los resultados que quiere conseguir.

En el ejemplo que se muestra a continuación se ve un script que muestra cómo modificar con filtros el tráfico generado por el servidor para 5 clientes definiendo las clases correspondientes con sus filtros. Se muestran únicamente los script de ejemplo, pues las gráficas del cambio de ancho de banda en los 5 clientes no aportarían mucha más información.

```

#!/bin/bash
DEV=eth0
RATE=10000
RATE1=9000
RATE2=2000
RATE3=5000
RATE4=6000
RATE5=1000
#CIASE I
sudo tc class add dev $DEV parent 1:1 classid 1:10 htb rate ${RATE1}kbit ceil ${RATE1}kbit prio 1
sudo tc qdisc add dev $DEV parent 1:10 handle 10: sfq perturb 10
#CIASE II
sudo tc class add dev $DEV parent 1:1 classid 1:20 htb rate ${RATE2}kbit ceil ${RATE2}kbit prio 2
sudo tc qdisc add dev $DEV parent 1:20 handle 20: sfq perturb 10
#CIASE III
sudo tc class add dev $DEV parent 1:1 classid 1:30 htb rate ${RATE3}kbit ceil ${RATE3}kbit prio 3
sudo tc qdisc add dev $DEV parent 1:30 handle 30: sfq perturb 10
#CIASE IV
sudo tc class add dev $DEV parent 1:1 classid 1:40 htb rate ${RATE4}kbit ceil ${RATE4}kbit prio 1
sudo tc qdisc add dev $DEV parent 1:40 handle 40: sfq perturb 10
#CIASE V
sudo tc class add dev $DEV parent 1:1 classid 1:40 htb rate ${RATE5}kbit ceil ${RATE4}kbit prio 1
sudo tc qdisc add dev $DEV parent 1:40 handle 40: sfq perturb 10
# FILTRO1
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip dst cliente1 flowid 1:10
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip src cliente1 flowid 1:10
# FILTRO2
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip dst cliente2 flowid 1:20
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip src cliente2 flowid 1:20
# FILTRO3
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip dst cliente3 flowid 1:30
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip src cliente3 flowid 1:30
# FILTRO4
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip dst cliente4 flowid 1:40
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip src cliente4 flowid 1:40
# FILTRO5
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip dst cliente5 flowid 1:50
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip src cliente5 flowid 1:50

```

También como ejemplo se muestra el script que se tendría que ejecutar en los clientes para modificar el tráfico que se envía al servidor en caso de querer emular la limitación del acceso de subida de cada cliente.

```

#!/bin/bash
DEV=eth0
RATE=10000
RATE1=9000
#CIASE I
sudo tc class add dev $DEV parent 1:1 classid 1:10 htb rate ${RATE1}kbit ceil ${RATE1}kbit prio 1
sudo tc qdisc add dev $DEV parent 1:10 handle 10: sfq perturb 10
# FILTRO1
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip dst servidor flowid 1:10
sudo tc filter add dev $DEV parent 1: protocol ip prio 1 u32 match ip src servidor flowid 1:10

```

3.3. Conclusiones

La herramienta es útil, procesa, captura y modifica el tráfico de red de la manera en la que se ha diseñado e implementado. Implementa el concepto de "Proyecto" para definir una estructura organizativa.

Es de fácil utilización, intuitiva y permite la automatización, pero para poder sacarle todo el potencial que la herramienta posee, es imprescindible que el investigador tenga los conocimientos avanzados suficientes en el uso de órdenes como: Tc y Netem y Tcpdump, y/o conocimientos avanzados de calidad de servicio y calidad de la experiencia, así como en redes de ordenadores o programación de script de Shell.

4. Cronograma

El presente documento constituye la Memoria de mi Trabajo Fin de Master (TFM), enmarcado en el Programa Oficial de Posgrado en Ingeniería de Sistemas e Informática, en la Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza. Está dirigido por el Dr. Enrique F. Torres Moreno del Departamento de Informática e Ingeniería de Sistemas, Área de Arquitectura y Tecnología de Computadores y por el Dr. Julián Fernández Navajas del Departamento de Ingeniería Electrónica y Comunicaciones, Área de Ingeniería Telemática y se enmarca dentro del grupo de investigación CeNITEQ (Communications Networks and Information Technologies For E-Health and Quality of Experience Group).

En el cronograma de la página siguiente se puede ver representado el tiempo que se invirtió en la realización de cada una de las partes de la que consta este trabajo fin de master.

1. Estudio y documentación para la realización del trabajo donde se recopiló información del grupo CeNITEQ y de distintos trabajos y estudios relacionados. Total 104 h.
2. Definición de la herramienta tras el estudio de un caso real: Se estudió en profundidad el juego Minecraft. Se tomaron las decisiones de diseño e implementación como realizar la herramienta en Bash o que aplicaciones se utilizarían, etc. Total 300 horas.
3. Desarrollo de la herramienta donde se implementó cada una de sus funciones siendo un proceso iterativo, teniendo que volver al punto anterior para redefinición de tareas y viceversa, realizando también pruebas parciales de cada parte diseñada. Total 644 horas.
4. Evaluación y testeo de la herramienta en distintos entornos reales y de laboratorio, con tráfico real y simulado. Se instalaron servidores y clientes de Minecraft, tanto en sistemas operativos Windows como Linux, se crearon escenarios o mapas con características especiales. Total 390 horas.
5. Elaboración de la memoria. Total 550 horas.

El total de horas invertidas en la realización del presente trabajo es de alrededor de 1.988 horas, divididas en un periodo de durante dos años y tres meses en el cual compatibilicé la realización del mismo con mi trabajo a tiempo completo.

5. Conclusiones

5.1. Conclusiones

En este trabajo se ha presentado la herramienta de control de tráfico de red para el análisis y evaluación de servicios multimedia interactivos, "Mastercraft", que facilita la ejecución de tareas técnicas que permiten ser automatizadas para la realización de experimentos.

La herramienta ha demostrado ser útil y funcionar adecuadamente realizando las tareas que se diseñaron para ella. Se ha puesto a prueba en entornos reales y en servicios multimedia interactivos de tiempo real. Se ha enfocado para los procedimientos técnicos que se realizan en el grupo de investigación de la Universidad de Zaragoza CeNITEQ.

Mastercraft se ha diseñado para que mejore la repetitividad de las pruebas o experimentos realizados, sea fácilmente modificable, sin necesidad de instalación e implementada en el intérprete de órdenes y lenguaje de programación de consola Bash de Linux.

Se implementó el concepto de "Proyecto" para definir la estructura organizativa creada por la herramienta que engloba todos los proyectos, capturas, escenarios y experimentos de una manera ordenada y coherente que facilita el trabajo.

La herramienta permite realizar capturas múltiples en equipos remotos facilitando la conexión a los mismos para su posterior análisis y procesamiento por medio de diferentes filtros automatizados. Dando información como por ejemplo, de cada una de las IP que aparecen en la captura y su puerto de conexión, de cuándo aparecen las mismas en la comunicación por primera vez y cuándo por última, el tiempo total de conexión, bytes, paquetes y ancho de banda instantáneo de cada uno de los elementos capturados, preparando así la información para su posterior análisis y procesamiento externo, en cualquier herramienta de procesamiento matemático de datos que el investigador utilice. En este trabajo se utilizó la herramienta Matlab para la presentación de resultados finales.

La herramienta permite gestionar el tráfico de red en una interfaz determinada, limitar el ancho de banda total disponible a un valor fijo conocido, limitarlo para un usuario con Tc o introducir retardos, pérdidas y duplicación de paquetes en la transmisión de paquetes de cualquier interfaz con Netem. Además de poder ir monitorizando los cambios realizados durante todo el proceso.

La facilidad en el uso de la herramienta por parte del investigador es una de las características principales del diseño y por ello Mastercraft permite realizar capturas de manera muy intuitiva. Lo mismo con el procesado, que es automático si se procesa por IP o puerto. Así mismo la generación automática de ficheros de ayuda para la preparación de las tablas, facilitan la realización de las mismas y estas tablas generan automáticamente los experimentos ejecutables que permiten modificar el tráfico de red. De esta manera se facilita el uso básico de la herramienta.

En el experimento 1B, al analizar los resultados proporcionados por la herramienta y pasados a gráficas con Matlab se ven diferentes patrones dependiendo si la captura realizada contenían acciones o movimiento dentro del juego o por el contrario se estaba quieto. La media de ancho de banda de subida para cada uno de los clientes es de aproximadamente 11 Kbps, y la del servidor de 186 kbps, una media muy baja. En el ejemplo de captura de tráfico de login y logout se ve un envío importante al principio de la conexión del servidor al cliente con paquetes de tamaño 400, después, el resto de la captura, paquetes del cliente al servidor de tamaño 44, que podrían indicar únicamente que el cliente está vivo.

Con estos experimentos 2A y 2B, se ha comprobado la variación del ancho de banda que se gestiona en la interface de red por medio de la herramienta desarrollada. Se ha generado tráfico en la red para posteriormente limitarlo con la herramienta y ver que lo realiza como se le especifica. Por una parte, se limitó el ancho de banda del enlace de salida del servidor y por otra se limitó el ancho de banda que reciben cada uno de los clientes con diferentes valores. De esta forma lo que se hace es emular el acceso de subida que tiene el servidor y los accesos de bajada que tienen los clientes.

Dicha herramienta ha sido presentada en el congreso, II Workshop QoS y QoE en Comunicación Multimedia (QOCM) que se celebró en Zaragoza en el mes de Julio de 2016 y está publicada para su libre utilización y consulta en la dirección: https://gitlab.unizar.es/dgimenez/Herramienta_Mastercraft que es un repositorio "gitlab" de gestión de proyectos y control de versiones que posee la Universidad de Zaragoza, además existen copias en posesión del grupo de investigación CeNITEQ y de los directores de este trabajo fin de master.

Otra conclusión a nivel personal es la cantidad de información que he adquirido sobre por ejemplo, generación de tráfico y gestión de interfaces de red, ancho de banda, perdidas y otros parámetros relacionados con la calidad de servicio. Aprender a trabajar dentro de un grupo de investigación en el área TIC.

5.2. Trabajos futuros

Un trabajo futuro sería aumentar aún más la utilidad de la herramienta, pudiendo modificar directamente el tráfico de red de una aplicación en concreto, en vez de modificar todo el tráfico. Para ello, habría que enviar cada uno de esos tráficos a una tarjeta de red virtual a la que pudiera atacar la herramienta y modificar su comportamiento.

Recientemente se ha presentado por parte de la empresa Microsoft la última versión de su sistema operativo, Windows 10, que integra una potente y mejorada consola Bash sacada del sistema operativo Linux Ubuntu. [MM16] [BMM16] Se tendría que ver hasta qué punto esto puede ser útil para la posible inclusión de la herramienta en los sistemas operativos Windows. Por poner un simple ejemplo que se tendría que analizar, conectarnos a dicha consola desde un sistema Linux donde se ejecutara la

herramienta "Mastercraft" y realizar un control remoto de ella. O ver que opciones tiene dicha consola para el control de las tarjetas de red en dicho equipo.

Otro trabajo futuro, hablado con el grupo, sería poder subir a la nube y poder compartir los resultados de los experimentos realizados con la herramienta.

Por último y pendiente de realización sería realizar un experimento completo, con un investigador sociólogo, un servidor, x jugadores y pasando después de la realización del experimento distintos *test* a los mismos para valorar la calidad de la experiencia en el juego después de realizar distintas modificaciones en el tráfico de red durante la partida, siguiendo los criterios de dicho investigador científico. Durante la realización de este trabajo, no ha existido esa posibilidad porque no ha surgido ningún proyecto, y queda pendiente para cuando surja.

6. Bibliografía

[DG16] David Giménez Muñoz; Julián Fernández Navajas; Guillermo Azuara Guillén; Idelkys Quintana Ramirez; José Ruiz Mas; José Luis Salazar Riaño; José María Saldaña Medina; Luis Sequeira Villarreal. "Configuración de entornos controlados de laboratorio para comunicaciones Multimedia" II Workshop QoS y QoE en Comunicación Multimedia (QQCM) Zaragoza, Spain, Julio 2016.

[IQ13] Idelkys Quintana, Jose Saldana, Jose Ruiz Mas, Luis Sequeira, Julián Fernández Navajas, Luis Casadesus, "Optimización del Tráfico P2P-TV mediante el uso de Técnicas de Compresión y Multiplexión," Jornadas de Ingeniería Telemática JITEL 2013, pp 345-350, Granada, Spain, October 28-30, 2013. ISBN-13: 978-84-616-5597-7.

[JS12] Jose Saldana, Julian Fernandez Navajas, Jose Ruiz Mas, Luis Sequeira, Luis Casadesus, "Comparison of Multiplexing Policies for FPS Games in terms of Subjective Quality". Proc. II Workshop on Multimedia Data Coding and Transmission 2012, Jornadas Sarteco. Elche (Spain). Sept. 2012. ISBN: 978-84-695-4472-3.

[QNT06] L. Stewart, P. Branch: Quake IV, Map: q4dm1, 5 players, 11Jan2006. Center for Advanced Internet Architectures SONG Database, http://caia.swin.edu.au/sitcrc/song/files/quake4_200706_1_q4dm1_5_fragment.tar.gz.

[LS12] Luis Sequeira, Julian Fernández-Navajas, Jose Saldana, Luis Casadesus, Jose Ruiz-Mas, "Empirically Characterizing the Buffer Behaviour of Real Devices," Proc. International Symposium on Performance Evaluation of Computer and Telecommunication Systems [SPECTS 2012](#), July 8-11, 2012, Genoa, Italy. ISBN: 978-1-4673-2235-5.

[CF14] Carlos Fernandez, Jose Saldana, Julian Fernandez-Navajas, Luis Sequeira, Luis Casadesus, "Video conferences through the Internet: How to Survive in a Hostile Environment," The Scientific World Journal, Vol. 2014.

[LS11] Luis E. Sequeira Villareal "Metodología para el modelado y el análisis de flujos Ip multimedia: Medidas de calidad" Director Julián Fernández Navajas Ponente José Ruiz Más Máster en Tecnologías de la Información y Comunicaciones en Redes Móviles. Programa oficial de Posgrado en Ingeniería de Telecomunicación Noviembre 2011.

[CF13] [Carlos Fernández Barberá "Análisis de prestaciones de un sistema de videoconferencia comercial" Proyecto fin de Carrera Ingeniería de Telecomunicación Especialidad Telemática, Director José María Saldaña Medina, Ponente Julián Fernández Navajas Julio 2013].

[IQ13b] Idelkys Quintana-Ramirez, Jose Saldana, Jose Ruiz-Mas, **Luis Sequeira**, Julian Fernandez-Navajas, Luis Casadesus, "Optimization of P2P-TV Traffic by Means of Header Compression and Multiplexing", [SoftCOM 2013](#), Split, Croatia, September 18-20, 2013. ISBN 978-953-290-041-5.

- [PRG15] Paessler <https://www.es.paessler.com/> Última visita 15/11/2015.
- [CNA15] Cisco <http://www.cisco.com/c/en/us/support/cloud-systems-management/prime-nam-2304-appliance/model.html> Última visita 15/11/2015. Última visita 15/11/2015.
- [CCA15] Cisco <http://www.cisco.com/c/en/us/products/interfaces-modules/catalyst-6500-series-network-analysis-module-nam-3/index.html> Última visita 15/11/2015.
- [ES12] Elisa Santos, Julián Fernández Navajas, Luis Sequeira, Luis Casadesus. "Herramienta para Automatizar la Caracterización de Entornos de Red: Análisis y Medidas de Calidad". Actas del XXVII Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2012). Elche (Spain). Sept. 2012. ISBN: 978-84-695-4327-6.
- [TD15] TCPDUMP & LIBPCAP <http://www.tcpdump.org/>. Última visita: 09/11/2015
- [WS15] <https://www.wireshark.org> . Última visita: 10/11/2015
- [JMS10a] J. M. Saldaña Medina, J. Murillo, J. Fernández Navajas, J. Ruiz Mas, E. A. Viruete Navarro, and J.I. Aznar Baranda. Emulación de escenarios de red mediante un testbed. *Actas del XXV Simposium Nacional de la Unión Científica Internacional de Radio (URSI)*, Bilbao (España). Septiembre 2010.
- [JMS10b] J. M. Saldaña Medina, J. Murillo, J. Fernández Navajas, J. Ruiz Mas, E. A. Viruete Navarro, and J.I. Aznar Baranda. Análisis de qos para una plataforma distribuida de telefonía ip. Actas de las IX Jornadas de Ingeniería Telemática (JITEL 2010). pages 63-70, Valladolid. Septiembre 2010.
- [JMS11a] J. M. Saldaña Medina, J. Murillo, J. Fernández Navajas, J. Ruiz Mas, E. A. Viruete Navarro, and J.I. Aznar Baranda. Qos and admission probability study for a sip-based central manager ip telephony system. *Proc. New Technologies and Security (NTMS), 5th International Conference*, Paris. ISBN: 978-1-4244-8704-2. Febrero 2011.
- [JMS11b] J. M. Saldaña Medina, J. Murillo, J. Fernández Navajas, J. Ruiz Mas, J. I. Aznar Baranda, Eduardo Viruete, and L. A. Casadesus Pazos. Influence of the router buffer on online games traffic multiplexing. *Proc. International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS*, pages 253-258, The Hague, Netherlands. ISBN: 978-161-782-309-1. Junio 2011.
- [MA03] Martin A. Brown: *Traffic Control HOWTO*, v1.0.1, Noviembre 2003.
- [MTC01] Man de Tc de Linux. lartc.org/mampages/tc.txt Última visita 10/11/2015.
- [IF15] Linux Foundation <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem> Última visita 10/11/2015.
- [NT11] Man de NETEM <http://man7.org/linux/man-pages/man8/tc-netem.8.html> Última visita 10/11/2015

[BOT12] A. Botta, A. Dainotti, A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios", *Computer Networks* (Elsevier), 2012, Volume 56, Issue 15, pp 3531-3547.

[ITG15] Universita' degli Studi di Napoli "Federico II" <http://traffic.comics.unina.it/software/ITG/index.php> Última visita 11/11/2015 .

[ETG11] L.A. Casadesus Pazos, J. Fernandez Navajas, J. Ruiz Mas, J.M. Saldaña Medina, J.I. Aznar Baranda, y E. Viruete Navarro, *Herramienta para automatización de medidas de tiempo real extremo a extremos*, Actas del XXVI Simposium Nacional de la Union científica Internacional de Radio (URSI 2011), Leganes (España). ISBN 9788493393458. Septiembre 2011.

[MAT15] MathWorks <http://es.mathworks.com/products/matlab/> Última visita 11/11/2015.

[CUM15] http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/garcia_b_s/capitulo2.pdf Última visita 11/11/2015.

[LSS15] <http://www.freeos.com/guides/lsst/> Última visita 11/11/2015.

[GN16] Gnuplot <http://gnuplot.sourceforge.net/> Última visita: 21/11/2016

[GNU15] GNU Operating System <https://www.gnu.org/software/bash/> Última visita 11/11/2015.

[GRY16] Awk <http://www.grymoire.com/Unix/Awk.html> Última visita: 25/10/2016

[AWK16] Gawk: Effective Awk Programming <http://www.gnu.org/software/gawk/manual/> Última visita: 25/10/2016

[IB16] <http://ibmonitor.sourceforge.net/> Última visita 04/11/2016

[IP16a] <https://iperf.fr/> Última visita 04/11/2016

[IP16b] <https://openmaniak.com/es/iperf.php> Última visita 04/11/2016

[MC15a] <https://minecraft.net/> Última visita 04/11/2015

[MC15b] <https://minecraft.net/stats> 21.310.006 copias Última visita 06/11/2015

[GZ15] <http://www.gamerzona.com/2015/07/01/minecraft-millones-uegos-vendidos/> Última visita 29/10/2015

[3DJ15] <http://www.3djuegos.com/noticia/144083/0/versiones-minecraft-consolas/superan-en-conjunto-ventas/pc/> Última visita 05/11/2015

[EM05] Eduardo Magaña, Edurne Izcue y Jesús Villadangos, *Análisis y Prestaciones de un Planificador de Tráfico sobre Plataformas de Propósito General*, Universidad Pública de Navarra.

https://www.researchgate.net/publication/228450449_Analisis_de_Prestaciones_de_un_Planificador_de_Trafico_sobre_Plataformas_de_Proposito_General Última visita
07/11/2016

Anexo 1. Estudio de los trabajos y metodología del grupo CeNITEQ

1. El grupo CeNITEQ

El grupo de investigación CeNITEQ (Communications Networks and Information Technologies For E-Health and Quality of Experience Group), pertenece a la Universidad de Zaragoza y su principal campo de trabajo son las nuevas Tecnologías de la Información y las Comunicaciones (TIC). Estas nuevas tecnologías permiten el desarrollo de la Sociedad de la Información a través de redes, aplicaciones y servicios que posibilitan la mejora en muchos aspectos de la vida actual. Con la aparición de nuevos servicios multimedia en tiempo real e interactivos como la videoconferencia, los juegos online, los sistemas de toma de decisiones y otros, se demandan altos niveles de calidad subjetiva por parte del usuario y por ello necesitan ser analizados para obtener, mantener, y mejorar en la medida de lo posible la percepción que el usuario tiene, y la manera en la que se le proporcionan dichos servicios (Quality of Experience, QoE).

Además el grupo CeNITEQ tiene tres líneas principales de investigación, que son: e-Health, QoE y las redes distribuidas. Este trabajo fin de master está centrado en línea de investigación relacionada con QoE, la cual esta soportada por un equipo de trabajo con apoyos multidisciplinares desde el ámbito social y que permite abordar las siguientes temáticas de investigación en los distintos proyectos en que se trabaja:

-Métodos de identificación de usuario en la toma de datos y toma de decisiones.

-Metodologías y herramientas para la mejora de las comunicaciones de red y la optimización de la calidad percibida.

En esta última área se trabaja en el desarrollo de métodos y herramientas que permitan adaptar las infraestructuras de red a los complejos patrones de tráfico que caracterizan a los nuevos servicios que despuntan, para conseguir una optimización real de la conectividad y para que el usuario también perciba una mejora en la calidad de los mismos.

Por este motivo, surge la necesidad de estudiar y buscar alguna herramienta enfocada al sector de la investigación, que brinde la posibilidad de analizar de forma automatizada y clara el comportamiento del tráfico de aplicaciones y servicios multimedia sobre un entorno de red, y que garantice la repetición y escalabilidad en su uso.

1.1. Estudio de los trabajos y metodologías

A continuación se analizarán diversos estudios que los investigadores de este grupo tienen dentro del área de telemática, y que están publicados en revistas científicas y congresos nacionales e internacionales, con esto se pretende estudiar dichos trabajos y poder extraer de ellos los procedimientos, las técnicas y/o las herramientas que utilizan y con todo ello poder presentar una propuesta de herramienta que pueda mejorar su trabajo

En [IQ13a], se puede ver como se aplica un método de compresión de cabeceras y unas políticas de multiplexación del tráfico de la aplicación *P2P-TV* llamada *SOPCast*, basada en UDP que tiene altas tasas de paquetes de pequeño tamaño, logrando un ahorro hasta de un 35% del ancho de banda de subida utilizado en redes residenciales,

así como, una reducción del número de paquetes por segundo. Se ha comprobado que dicho proceso de multiplexión no perjudica la experiencia del usuario con la aplicación ni la calidad del video percibido por el mismo. Para ello se utilizó el tráfico de la propia aplicación de *SOPCast* y se capturo con *Tcpdump*. El tráfico de red capturado era únicamente UDP de un puerto específico, se utilizaron scripts programados en *shell* de Linux para el procesamiento de datos y Matlab para el análisis y generación de gráficas.

En [JS12] se comparan 2 políticas para multiplexar el tráfico generado por jugadores de juegos online tipo *First Person Shooter* (FPS), se analizan los resultados de calidad subjetiva que el jugador percibe. En la primera política se ahorra en ancho de banda, y con la segunda se tienen menos retardos y jitter en la red. Con pocos jugadores se tendría que utilizar la primera política y con muchos jugadores la segunda para la comparación de dichas políticas utilizan las capturas de [QNT06], donde vemos que el juego analizado “Quake4” el cual está basado en juegos tipo cliente-servidor, que envía únicamente tráfico UDP y que las capturas se realizan con *Tcpdump*.

En [LS12] se presenta un procedimiento para determinar las características técnicas y funciones de los *buffer* de los *routers*. Utiliza dos métodos distintos para el análisis de los *routers* con indiferencia de si se tiene acceso físico al *router* como si no se tiene dicho acceso, para ello utiliza un generador de tráfico creado por miembros del propio grupo, las capturas se realizan con *Tcpdump* y se utiliza para poder además realizar capturas remotas. El propio investigador durante cada experimento realiza manualmente las conexiones a los equipos remotos para cada captura. Se crean script-Shell de Linux para el análisis del tráfico UDP y para realizar cálculos de ancho de banda, pérdidas, etc utiliza la herramienta *Tc*. Se utilizó Matlab para el procesado de cálculos y generación de gráficas.

En [CF14] se comparan dos soluciones de videoconferencia una tipo cliente-servidor y otra tipo peer-to-peer con el objetivo de analizar su comportamiento cuando las características de la red por la cual transitan los datos cambia, para ello se ha construido en un laboratorio, un entorno de red con diferentes ordenadores capaces de introducir artificialmente limitaciones de ancho de banda, pérdida de paquetes y retardos en la red. Genera tráfico UDP simulado con un generador de tráfico de red que después es capturado utilizando *Tcpdump* y para la gestión del tráfico utiliza la herramienta *Tc*.

Los siguientes trabajos van a permitir obtener y entender el problema en detalle, al describir las metodologías que realizan los investigadores del grupo.

En [LS11] se propone una metodología para el análisis y modelado de flujos IP de datos multimedia a través de las redes Wifi. El trabajo quiere obtener distintos modelos de tráfico multimedia, y generar tráfico en entornos de laboratorio que después se puedan extrapolar a entornos reales, automatizando el proceso y ampliándolo a otros tipos de tráfico.

En [CF13] se analiza la aplicación de videoconferencia “*Vidyo*” y la aplicación *Skype*, realizando una instalación en diferentes sistemas operativos y en diferentes escenarios de red, dentro de un entorno controlado de laboratorio, después se han introducido cambios en las condiciones de red como limitaciones de ancho de banda,

pérdidas y retardos y se ha analizado los resultados obtenidos, extrayendo conclusiones sobre las características y el funcionamiento de ambas soluciones de videoconferencia.

Y por último, en [IQ13-b] se muestra una optimización del tráfico de una aplicación P2P-TV (aplicaciones multimedia en tiempo real), que genera paquetes pequeños UDP a altas tasas de transferencia, generando redundancia y un *overhead* en las redes, debido a los campos repetidos de las cabeceras. Se propone un método de optimización del ancho de banda, basándose en primer lugar en la compresión de las cabeceras y posteriormente en la multiplexión de los paquetes originales. Se han comparado dos técnicas diferentes, la primera tomando un periodo fijo y en la segunda se tiene en cuenta un umbral para el tiempo entre paquetes. Se obtienen en ambas políticas una mejora en la eficiencia y valores significativos del ancho de banda en el enlace de subida (upload). Además, el número de paquetes por segundo se reduce en un factor de 10 aproximadamente. Sin embargo, se añaden retardos a los paquetes nativos, aunque esto no empeora la experiencia del usuario ni la calidad del vídeo percibido, debido a los mecanismos de ajuste de la aplicación P2P.

La metodología de estos trabajos se puede fundir en los siguientes 5 pasos:

1. Definir el escenario
2. Configurar el entorno físico
3. Realizar experimento
4. Comprobar experimento
5. Presentación de resultados

En la primera fase en [LS11] es el estudio teórico y la planificación de la prueba, recopilar la información de las fuentes necesarias, analizar dicha información y seleccionar las aplicaciones a modelar, el equipamiento a utilizar además de elegir las herramientas para la obtención de datos y para el análisis de resultados. Finalmente, asignar las distintas tareas a grupos de trabajo y elaborar un cronograma de actividades. En [CF13] es el diseño de un esquema de red para la realización de las pruebas con escenarios simulados, similares a distintas soluciones de videoconferencia reales, sólo se genera tráfico de vídeo en un sentido. El escenario 1, es el de emisión, donde se estudia el enlace de subida y el escenario 2 es el de recepción, donde se analiza el enlace de bajada. Y en [IQ13-b] es un análisis teórico y caracterización del tráfico de, centrándonos en el tamaño de los paquetes y sus funciones.

La segunda fase en [LS11] es el acondicionamiento del entorno de pruebas, el cual debe estar libre de ruido y de cualquier interferencia, también hay que realizar y comprobar las conexiones físicas de la red y poner en marcha los equipos. Seguidamente configurar los parámetros básicos y comprobar los enlaces de la red y monitorizar los procesos activos y los recursos, finalmente se eliminaran los procesos innecesarios en el sistema. En [CF13] se sitúan entre los usuarios y el servidor algunos elementos de red que facilitan la realización de las pruebas: *Router*, *hub* y equipo auxiliar para simular diferentes condiciones y limitaciones en la red. Y en [IQ13-b] es el montaje de un escenario de pruebas, donde el equipo cliente de *SopCast* se encuentre en la red de nuestro campus universitario. Nuestro cliente forma parte de una red *P2P-TV* y comparte el mismo canal de TV con otros peers ubicados en Internet. Este escenario debe quedar libre de interferencias y ruidos, se debe aislar el tráfico de la aplicación

P2P-TV (SopCast) de otros posibles tráficos que puedan estar generándose en nuestro equipo.

La tercera fase en [LS11] es la obtención de resultados lanzando y configurando las aplicaciones y las herramientas para la obtención de resultados y ejecutar la herramienta para la captura de datos. Hecho esto, iniciar la transmisión manteniéndola activa el tiempo planificado y al acabar este, apagar la transmisión y parar la herramienta de obtención de resultados. En [CF13] es la selección de las herramientas y las aplicaciones para desarrollar las pruebas necesarias con el fin de caracterizar el funcionamiento en los escenarios descritos. Las herramientas son el Proxy ARP, *Traffic Control* (Tc) y *Tcpdump*. Y en [IQ13-b] una vez que se tiene el escenario de pruebas montado, se lanza la herramienta *Tcpdump* (en este caso por la interfaz de red que da salida a Internet) con el fin de capturar el tráfico generado por *SopCast*, durante un tiempo de 90 minutos aproximadamente.

La cuarta fase en [LS11] es el análisis de resultados para lo cual se tiene que verificar el contenido de los resultados obtenidos, construir las herramientas o los procedimientos que permitan adaptar los resultados y que sean reconocidos como parámetros de entrada en las herramientas de análisis. Para ello se necesita exportar dichos resultados a formatos compatibles con las aplicaciones de análisis, aplicar dichas herramientas y verificar la correspondencia de los resultados obtenidos cotejándolos con los resultados de los trabajos de otros investigadores. En [CF13] es la captura del tráfico en las interfaces de red indicadas y su posterior almacenamiento en ficheros. Y en [IQ13-b] En la cuarta fase, se lanza la aplicación *P2P-TV* en el equipo cliente (se obtienen todos aquellos paquetes de señalización, control y también de datos).

La quinta y última fase en [LS11] es la presentación de resultados que se realiza seleccionando las herramientas de elaboración de documentos, gráficos, figuras y síntesis de resultados conseguidos para la posterior producción de documentos científicos. En [CF13] se ve el procesamiento de los ficheros que contienen las capturas para que mediante unos parámetros y reglas se obtenga un tráfico útil en este estudio. Y en [IQ13-b] se procede a analizar mediante scripts (en *Shell*) los datos obtenidos en la captura, separando el tamaño de los paquetes, el tiempo de transmisión (o el tiempo entre paquetes), las direcciones IP origen y destino, el puerto y filtrando por el protocolo UDP. Con el fin de analizar los resultados y adaptarlos a un formato de entrada compatible con la herramienta *Matlab*. Donde se procesan y se obtienen los resultados mostrados en este trabajo (ahorro de ancho de banda, número de paquetes por segundo, tiempo entre paquetes, etc.)

Como se ha podido ver, los investigadores pasan mucho tiempo aprendiendo el manejo de varias herramientas, como por ejemplo *Tcpdump* para la captura de datos, al igual que *Tc* y *Netem* para la gestión del tráfico de red, o el *Shell* de Linux para la ejecución de scripts y *Matlab* para la creación de gráficos. La ejecución de estas no está automatizada de ninguna manera y al repetir los experimentos se cometen frecuentes errores y además, tienen que configurar múltiples parámetros para que se puedan ajustar a los requerimientos específicos de los distintos escenarios o experimentos con la considerable pérdida de tiempo que esto supone.

Además también dedican tiempo a desarrollar e implementar ellos mismos las herramientas específicas que necesiten, como por ejemplo un generador de tráfico de red, que le permita realizar las requeridas mediciones y/o modificaciones de tráfico de red, para poder finalizar en un tiempo razonable sus proyectos. Todo el tiempo que se invierte en la búsqueda de herramientas o en su desarrollo sería mejor poder invertirlo en su propia investigación.

Todos estos trabajos tienen en común además de las herramientas mencionadas anteriormente, la implementación, normalmente dentro de un laboratorio de escenarios de red, en los cuales hay que configurar, aplicaciones, redes, equipos, etc. Y todo ello también lleva una considerable cantidad de tiempo y recursos para poder desarrollar dichos estudios.

Es por ello, que sería muy útil disponer de una herramienta que aglutinara las anteriormente mencionadas, o que realice sus mismas funciones y además pudiera automatizar ciertos procesos comunes de todos estos trabajos mencionados.

HERRAMIENTA DE CONTROL DE TRÁFICO DE RED PARA EL ANÁLISIS Y EVALUACIÓN DE SERVICIOS MULTIMEDIA INTERACTIVOS.

Esta es una guía de referencia para cualquier usuario que utilice la aplicación que he realizado para el Trabajo Fin de Master (TFM), enmarcado en el Programa Oficial de Posgrado en Ingeniería de Sistemas e Informática, en la Escuela de Ingeniería y Arquitectura de la Universidad de Zaragoza.

La guía se compone de los siguientes capítulos:

1. Instalación
2. Creación de un proyecto
3. Menú principal
4. La captura de tráfico de red
5. El procesado del tráfico de red
6. Compartir claves
7. Preparar Experimento
8. Generar Experimento
9. Lanzar Experimento
10. Ejecutar comandos
11. Cambio de Proyecto

1. Instalación

La herramienta de control de tráfico de red para el análisis y evaluación de servicios multimedia interactivos está publicada para su libre uso en un repositorio “gitlab” de gestión de proyectos y control de versiones que posee la Universidad de Zaragoza, <https://gitlab.unizar.es/dgimenez/Mastercraft.git>, además existen copias en posesión del grupo de investigación CeNITEQ y de los directores de este trabajo fin de master.

Para realizar la instalación del programa, solo tenemos que descomprimir el fichero del programa en la carpeta se quiera, una vez que se tenga descomprimido, se buscará el fichero llamado Mastercraft (pudiéndose cambiar el nombre del programa si se desea) y se garantizará que posee permisos de ejecución. (*chmod +x Mastercraft*).

Después de asegurarse de que el fichero de ejecución del programa tiene permisos de escritura, se abrirá una consola, se llegara hasta la carpeta en la que se ha descomprimido y se ejecutaremos de la siguiente forma: (*./Mastercraft*)

El programa detecta el sistema operativo Linux en el que se está trabajando y lanza la consola correspondiente, así es compatible con cualquier Shell y versión del sistema operativo. (Es recomendable que se amplíe un poco el tamaño, tanto en altura como en anchura, de pantalla de la consola para que así se pueda mostrar completamente las tablas o estadísticas que generen los proyectos).

2. Creación de un proyecto

Al entrar en el programa se muestra la ruta completa donde está instalado nuestro programa y se pregunta "Nombre de proyecto:". Aquí se tiene que indicar el nombre del proyecto de investigación donde se realizan las capturas (Se pueden crear cuantos proyectos se quieran). Es un buen método para tener todas las capturas ordenadas por proyecto o por nombre de investigador, así todos los investigadores pueden utilizar la herramienta en el mismo equipo, indicando cada investigador, cuando entre al programa, el nombre de su o sus proyectos. Podemos verlo en la figura 1.1.

```
[david@localhost Mastercraft]$ ./Mastercraft
Raiz: /home/david/Pruebas/Mastercraft

Nombre de proyecto: █
```

Figura 1.1 Nombre de Proyecto

Al introducir el nombre del proyecto, si este existe ya, se entra en el para seguir trabajando.

Si el proyecto no existe, porque es nuevo o porque es la primera vez que utilizamos el programa, se entra en una pantalla como se ve en la figura 1.2, donde se permite si se necesita, instalar las herramientas que el programa utiliza o puede utilizar para desempeñar sus tareas.

```
[david@localhost Mastercraft]$ ./Mastercraft
Raiz: /home/david/Pruebas/Mastercraft

Nombre de proyecto: Manual

Si es la primera vez que utilizas este programa en este ordenador
verifica que tienes todo el software que necesitas

Las dependencias que se van a utilizar son las siguientes

tcpdump nano gnuplot

1) Red Hat, fedora u otras con gestor de paquetes YUM
2) Debian, Ubuntu, u otras con gestor de paquetes APT-GET
3) Suse, OpenSuse, u otras con gestor de paquetes ZIPPER
4) Archlinux, Chakra u otras con gestor de paquetes PACMAN
5) Salir

Seleccionar número [1-5] █
```

Figura 1.2 Nuevo Proyecto

Se nos permite realizar una instalación automática indicándole únicamente el sistema operativo Linux que se utilice. En el momento de la preparación de esta guía de uso son: Tcpcdump, Nano y Gnuplot. Al finalizar, pulsando el 5 se sale directamente al menú principal de la aplicación.

Se da por descontado que se ha instalado tanto en el equipo local donde se ejecuta esta aplicación como en los equipos Linux remotos donde se tendrá que acceder para nuestros experimentos, todas las utilidades y herramientas básicas y típicas de cualquier sistema Linux, como por ejemplo el programa “sudo” (instalado por defecto) que es una utilidad que permite a los usuarios ejecutar programas con privilegios de seguridad de otro usuario (normalmente root).

Cuando se crea un proyecto nuevo el programa genera una carpeta con el nombre del proyecto, dentro de la carpeta “Proyectos” y 3 subcarpetas, llamadas, “Capturas”, “Procesar” y “Tablas” donde nos irá ordenando cada uno de los ficheros que se generan, como podemos ver en la siguiente imagen.

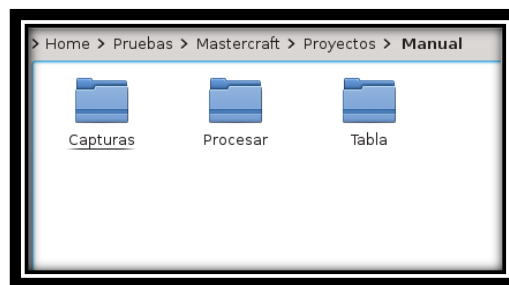


Figura 1.3. Carpetas del proyecto

3. Menú principal

Esta es la pantalla del menú principal:

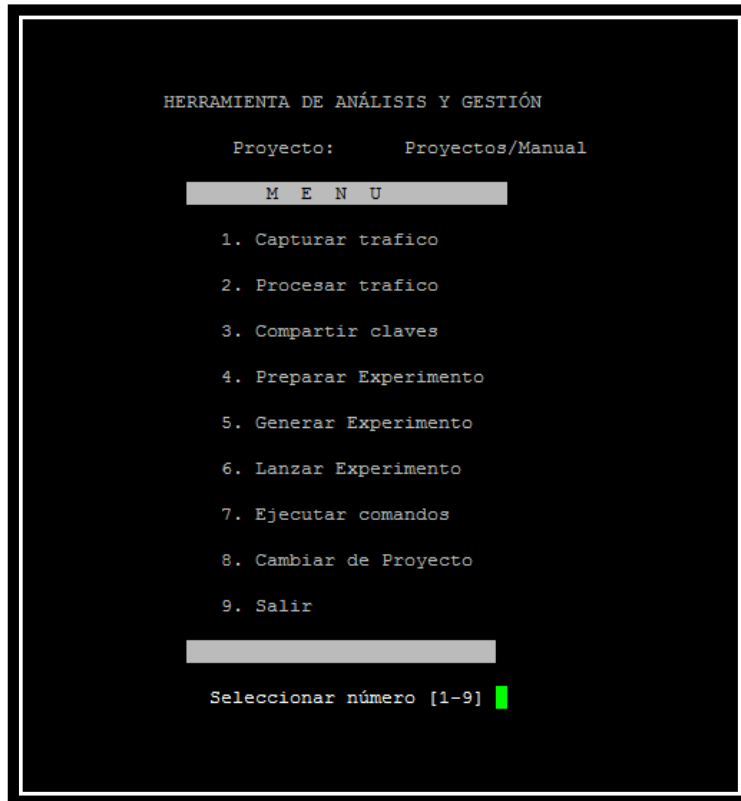


Figura 1.4 Menú Principal

Esta pantalla es la más importante y a la que se vuelve cada vez que se haya acabado cada uno de los procesos dentro de la aplicación. Nos indicara siempre, el nombre el proyecto en el que se está trabajando.

4. La captura de tráfico de red

Una de las partes importantes del programa es la captura de tráfico, cuando se elige la primera opción del programa "Captura de tráfico" como podemos ver en la imagen 1.5, nos muestra un submenú en el que se permite elegir entre una captura local o una captura remota, además de la opción salir, que se devolvería al menú principal de la aplicación.

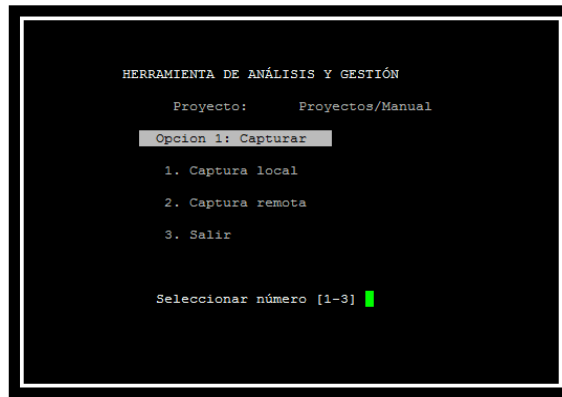


Figura 1.5 Menú Captura

La opción “1. Captura local” numera y muestra por pantalla, todas las interfaces de red que se tienen en el ordenador local donde se ejecuta Mastercraf, tanto redes cableadas como inalámbricas y se permite elegir a través de cuál de ellas quiera realizar la captura de trafico de red. (Es posible que se necesite en determinadas opciones del programa validar con permisos de administrador, se tendrá que introducirle la contraseña de root del equipo).

Una vez elegida la interfaz de red por la que se va a capturar, se pide un nombre para guardar dicha captura, este nombre es el que utilizará como nombre del fichero “.pcap” donde guardará la captura, también nos pedirá que introduzcamos tiempo de captura en segundos. En la captura que se muestra en la figura 1.6 se le da el nombre captura1 y 60 segundos de tiempo de captura.



Figura 1.6 Captura local 1

Pasados los 60 segundos, en la carteta genérica de “Proyectos”, dentro de nuestro proyecto, en este caso “Manual” y en la subcarpeta “Capturas” aparecerá el fichero con la captura realizada. “Captura1.pcap” como podemos ver en la figura 1.7.

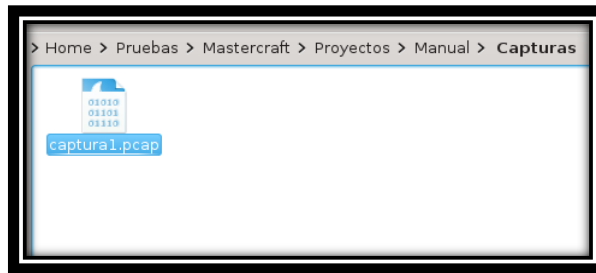


Figura 1.7 Captura local 2

La opción “2. Captura remota” permite realizar capturas de red en equipos Linux remotos, (Antes de realizar una captura remota ver apartado “6. Compartir claves”, en el cual se realiza una conexión Ssh con el equipo remoto para que no tenga que pedir contraseñas durante la captura). Lo primero que nos pide el programa son los datos de conexión, que consisten en el nombre de usuario remoto, una “@” (arroba), seguido de la IP o host remoto, una “%” (tanto por ciento) y la interface de red del equipo remoto donde se quiere realizar la captura. Esto se quiere para cada uno de los equipos remotos en los que se quiera realizar una captura separados los por un espacio y después pulsamos “Enter”. (En la figura 1.8 utilizaremos: cliente1@10.3.12.98%eth0). Después introducimos, igual que en el caso de la captura local, el nombre de la captura y el tiempo de captura.

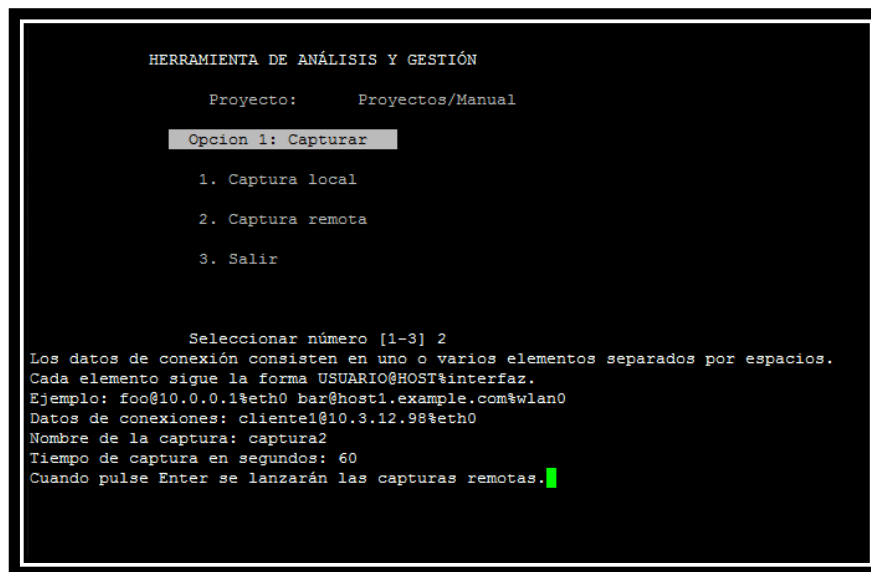


Figura 1.8 Captura remota 1

Igual que en la captura local, pasados los 60 segundos, (más unos segundos que añade el programa para asegurar que se cierra bien la captura antes de traerla al equipo local) en la carteta genérica de “Proyectos”, dentro de nuestro proyecto, en este caso “Manual” y en la subcarpeta “Capturas” aparecerá el fichero con la captura realizada. “Captura2_10.3.12.98_eth0.pcap”. El nombre se compone del nombre de captura, guión bajo, IP remota, guión bajo, dispositivo de red. Así se asegura el poder hacer capturas en el mismo equipo remoto en tarjetas de red distintas, por ejemplo Ethernet y wifi y que las guarde con distinto nombre.

Se traen al equipo local una copia del fichero de captura, como se ve en la figura 1.9, porque es en este equipo donde se procesa todas las capturas, quedándose en el

equipo remoto en su carpeta “/tmp” la captura original, que permanecerá allí hasta que se reinicie el equipo remoto.

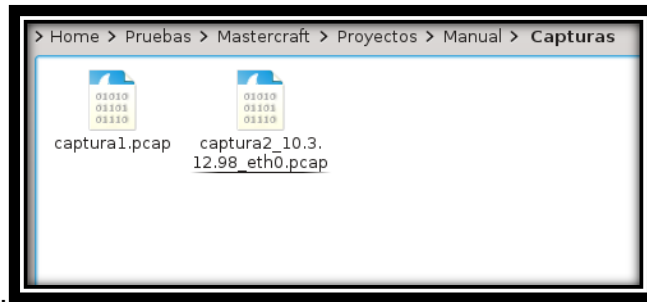


Figura 1.9 Donde se ve una captura local y una remota

5. El procesado del tráfico

La segunda opción del menú, ofrece el procesamiento de tráfico de red de las capturas que se tengan almacenadas en la carpeta “Capturas” de nuestro proyecto.

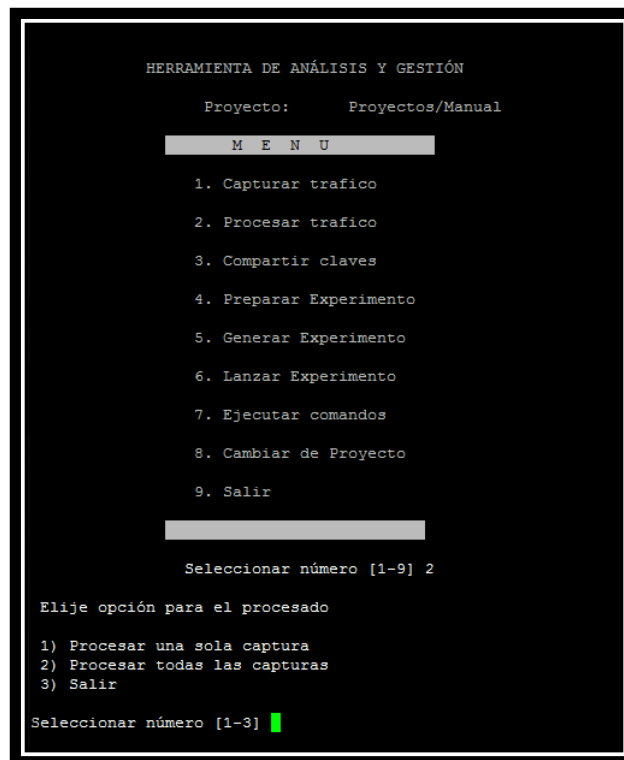


Figura 1.10 Procesar trafico

Como se ven en la figura 1.10, se permite procesar una sola captura o todas las capturas, la primera opción “una sola captura” procesa y saca resultados y estadísticas de una sola captura, por ejemplo de un servidor o una subred local o un laboratorio donde se tengan en una sola captura todos los datos necesarios, o si simplemente sólo se quiere revisar una sola captura. La segunda opción “todas las capturas” permite, hacer capturas remotas y/o locales y poder verlas todas, una detrás de otra y poder analizar los resultados de una manera más sencilla. Por

ejemplo se tienen capturas en un servidor y de 5 clientes, se puede ver primero sola la captura del servidor y después ver las de todos los clientes juntas, una detrás de otra.

```
Elige opción para el procesado
1) Procesar una sola captura
2) Procesar todas las capturas
3) Salir

Seleccionar número [1-3] 1
Qué captura quieres procesar?:
1 captura1.pcap
2 captura2_10.3.12.98_eth0.pcap

Digita el número de la captura: 1
Elige opción para filtrar las capturas

1) Filtrado por puerto
2) Filtrado por Ip
3) Filtrado Manual
4) Salir

Seleccionar número [1-4] █
```

Figura 1.11 Una sola captura

Como se ve en la figura 1.11, se ofrecen tres opciones para el filtrado de la captura. Filtrado por puerto, filtrado por IP y filtrado Manual.

En la opción “1” que suele ser la más común, se puede decir que se filtre por el puerto que se quiera o necesite.

En el caso de las capturas que se están poniendo como ejemplo en la figura 1.12, será el puerto 5050, que es el que utiliza el servidor de Minecraft para enviar y recibir paquetes de datos por la red. Se indica que “Este proceso puede tardar varios segundos dependiendo del tamaño del fichero que se le pase a procesar.

```
Seleccionar número [1-4] 1
Procesando captura...

Este proceso puede tardar varios segundos
Dame puerto: 5050█
```

Figura 1.12 Captura por puerto

En la carpeta “capturas” el procesamiento va generando los siguientes ficheros:

- Uno con todas las IP’s capturadas con sus puertos.
- Cada una de las IP’s con su tráfico de subida “up”.
- Grafica del ancho de banda instantáneo por IP de subida.
- Cada una de las IP’s con su tráfico de bajada “down”.
- Grafica del ancho de banda instantáneo por IP de bajada.

Y para que los investigadores no tengan que repasar uno por uno todos estos archivos generados, el programa muestra por pantalla de consola un resumen de todo

lo que hemos procesado. Indicando que está haciendo en cada momento mientras procesa la información.

Se indica que está descubriendo las IP de la captura, que está generando los ficheros de trafico de red de subida y de bajada, se indica el tiempo total de la captura, el número total de paquetes capturados, así como una completa y estructurada tabla con toda la información que el investigador puede analizar de un solo vistazo y así poder acceder a los ficheros específicos con mayor precisión y velocidad, la tabla que se muestra, contiene los siguientes campos:

- IP de captura con su puerto
- Inicio de sesión por IP y puerto
- Fin de sesión por IP y puerto
- Tiempo total de la sesión
- Total bytes de subida por IP y puerto
- Total bytes de bajada por IP y puerto
- Total de paquetes de subida por IP y puerto
- Total de paquetes de bajada por IP y puerto
- Ancho de banda de subida por IP y puerto
- Ancho de banda de bajada por IP y puerto

También se indica cuando genera los gráficos de ancho de banda instantáneo de las IP tanto de subida como de bajada, utilizando para ello dentro de nuestro script el comando Gnuplot. Gnuplot es un pequeño programa en línea de comandos que permite dibujar gráficos usando una tabla de coordenadas (en formato sólo texto) que se le pasa por cada una de las IP.

```

Descubriendo IP's de las capturas
Generando ficheros de Up y de Down

Extrayendo estadísticas de las capturas

Generando ficheros de resumen de las capturas

Estadísticas de la captura: captura1.pcap_total.txt_final.tmp

Tiempo total de captura: 320731949 microsegundos

Nº total de paquetes capturados: 41020

|Nº| Ip_y_Puerto | Inicio_Sesion | Fin_sesion | Sesion | Bytes_Up | Bytes_Down |Packet_up|Packet_down|AB_up|AB_down|
-----
1 155.210.140.139.13229 1430140821418699 1430140821422901 4202 2417 757 60 18 225.324 160.533
2 155.210.140.139.13230 1430140821418978 1430140821460917 41939 264 247 6 5 53.6053 64.262
3 155.210.140.139.13232 1430140822315861 1430140961570118 139254257 201150 2181665 3332 4138 1.55092 24.2322
4 155.210.140.160.56235 1430140824193558 1430140824462160 268602 304 287 7 6 45.8275 53.5402
5 155.210.140.160.56239 1430140824865078 1430140984899177 160034099 299964 3175340 4910 5709 10.4856 27.4893
6 155.210.140.160.56495 1430140986105278 1430140995099057 8993779 120 0 4 0 80 0
7 155.210.140.8.50684 1430140734028216 1430140734059103 30887 2457 367 61 8 180.739 60.2493
8 155.210.140.8.50685 1430140734028321 1430140734034568 6247 1580 717 39 17 168.057 120.381
9 155.210.140.8.50686 1430140736636597 1430140971868745 235232148 317032 3123772 5626 6399 2.01424 20.0389
10 155.210.140.8.59753 1430140975774394 1430140975808778 34384 2178 277 54 6 233.407 53.6224
11 155.210.140.8.59755 1430140975774502 1430140975777984 3482 264 277 6 6 53.9307 71.3546
12 155.210.164.42.51741 1430140726266439 1430140726320970 54531 2456 327 61 7 242.692 91.6095
13 155.210.164.42.51742 1430140726266833 1430140726270283 3450 2456 757 61 18 225.646 166.607
14 155.210.164.42.51746 1430140749146217 1430140951761030 202614813 273223 2702606 4831 5574 4.42242 19.6947
15 155.210.164.42.51752 1430141046994864 1430141046994864 0 40 0 2 0 160 0
16 155.210.164.42.51753 1430141046994866 1430141046998388 3522 2497 717 62 17 235.251 179.404
17 155.210.164.46.5050 1430140726266439 1430141046994864 320728425 1118488 1099188 21844 18884 30.5967 8.25048
18 155.210.164.48.5050 1430140726266833 1430141046998388 320731555 3225 9214 72 224 140.736 211.907

Procesando graficos de captura

...Fin del procesado

Pulsa ENTER para volver al Menu: █

```

Figura 1.13 Estadísticas Resultados

Finalmente se indica que ha terminado el procesado y se mantiene la información en pantalla para que se pueda analizar hasta que se pulse la tecla “Enter”. En ese momento borra todos los datos temporales almacenados en la carpeta “Capturas” y que se ha generado con el procesamiento, dejando únicamente las capturas que ya se tenían (los “.pcap”) y todos los demás archivos generados en el procesado, se pasan automáticamente a la carpeta “Procesar” como vemos en la figura 1.14 y dentro de esta a una subcarpeta con el nombre de la captura, para que podamos identificarla posteriormente, donde se quedan definitivamente para que el investigador pueda hacer un análisis más profundo de los datos, incluidos los datos y tablas que han aparecido en la pantalla.

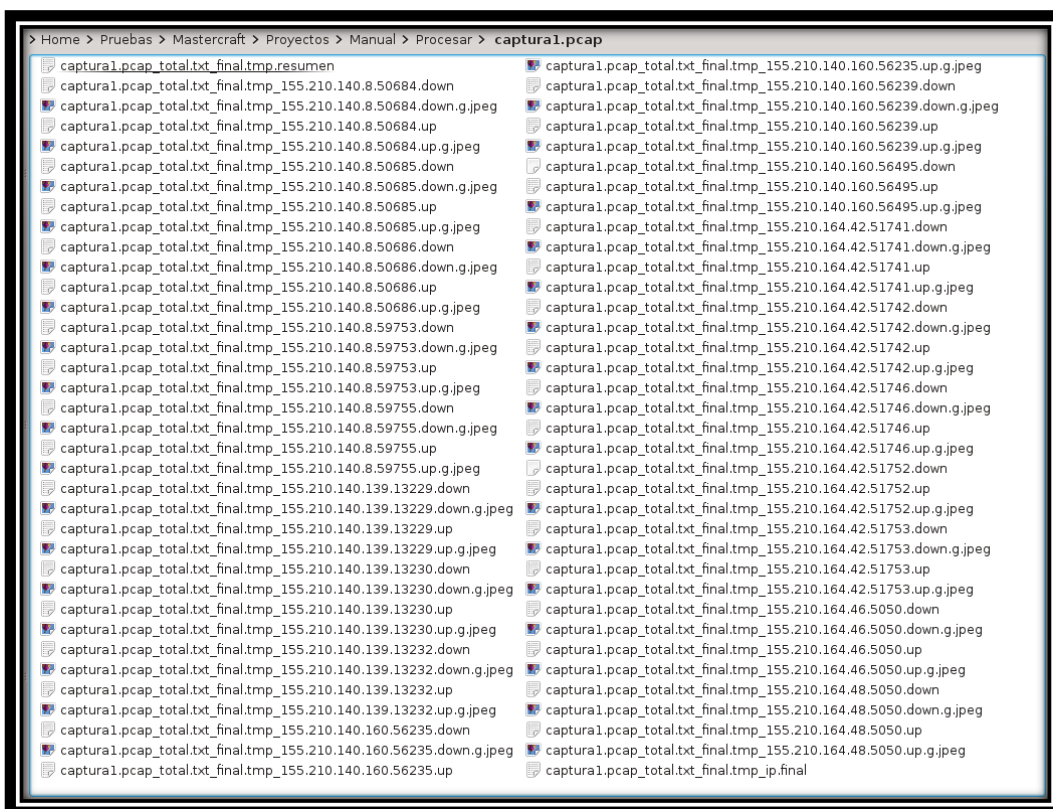


Figura 1.14 ficheros procesados

En la opción “2” del procesado de tráfico de red, es “filtrado por IP” en el que únicamente se tiene que dar una IP y el realizada los pasos indicados anteriormente, pero solo centrándose en la IP que se ha indicado. La tabla de estadísticas que sacará por pantalla dará información sobre esa IP, sea esta el origen del tráfico o el destino, con todos los bytes y paquetes que envíe o reciba, además del ancho de banda instantáneo que genere.

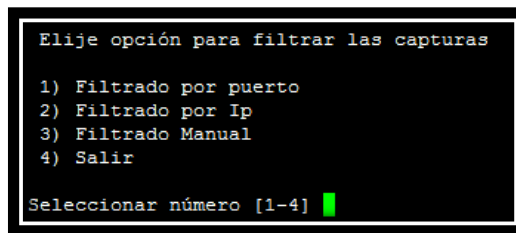


Figura 1.15 Puerto, Ip o Manual

La opción “3” del procesado de tráfico de red, es “filtrado manual”, para que se pueda pasar al comando de procesamiento (Tcpdump) los parámetros que se necesiten y que no sean puerto o IP. Como por ejemplo el procesado del tráfico TCP, UDP, ICMP, ARP, etc. Con el parámetro “src” y una IP filtrará el tráfico que tuviera su origen en dicha IP y con “dst”, que tuviera esa IP por destino. Con “not” más un tipo de tráfico como UDP se procesará todo el tráfico menos el UDP. Un ejemplo más complicado, pero que da una idea de la potencia de esta opción sería, hacer un procesado de los paquetes de inicio y fin (SYN y FIN) de cada sesión TCP y que no involucre como destino a ninguna red local, lo que se tendría que poner sería lo siguiente:

```
'tcp[tcpflags] & (tcp-syn|tcp-fin) != 0 and not src and dst net localnet'
```

Para más información sobre esta última opción del procesado, se recomienda revisar la página man de Tcpdump: (`$ man tcpdump`).

6. Compartir claves

Esta tercera opción del menú: “Compartir claves”, permite conectar a cualquier equipo con sistema operativo Linux de manera segura a través del algoritmo RSA. Este es un algoritmo asimétrico que cifra en bloques, utiliza una clave pública, la cual es la que distribuye y otra clave privada, que es la que guarda en secreto el propietario, obteniendo de esta forma una conexión completamente segura.

El programa pide una IP de destino y un usuario para conectarse. Si el equipo remoto permite o tiene habilitada la conexión con el usuario administrador “root”, se puede conectar con dicho usuario y ya no se tendría que hacer nada más para las capturas remotas o para lanzar los experimentos de modificación de tráfico de red. Por políticas de seguridad, el usuario “root” no suele estar habilitado para conexiones remotas. En ese caso se tendría que validar con un usuario normal del equipo remoto al que habría habilitado permisos de ejecución concretos y así poder ejecutar los programas necesarios en el equipo remoto.

Se usará un usuario ya creado en el equipo remoto o se creará un usuario propio para nuestros experimentos en cada una de las máquinas remotas a las que se quiere conectar. Desde dicha máquina remota se ejecutará el comando “visudo” con permisos de administrador, “sudo visudo” que abrirá el fichero sudoers al cual se añadirá al final del mismo las siguientes 3 líneas.

```
#config for mastercraft without password  
Cmnd_Alias CMD_MASTERCRAFT = /usr/sbin/tcpdump, /usr/sbin/tc, /tmp
```

david ALL = (ALL) NOPASSWD:CMD_MASTERCRAFT

```
#config for mastercraft without password
Cmdnd Alias CMD_MASTERCRAFT = /usr/sbin/tcpdump, /usr/sbin/tc, /tmp
david ALL = (ALL) NOPASSWD:CMD_MASTERCRAFT
```

Figura 1.16 sudoers

Tal como se ha indicado en el ejemplo, al usuario “david” se le da permisos de ejecución sin que tenga que pedir contraseña (aunque sigue siendo necesario poner sudo antes del programa a ejecutar) a los programas “Tcpdump”, “Tc” y a la carpeta temporal “tmp”, se tiene que asegurar que los programas que se quieren ejecutar estén en la ruta “/usr/sbin/”, si no es así, únicamente se tendría que cambiar la ruta. Así mismo, si en el fichero sudoers, el cual se edita con visudo que aparece en la línea “default requiretty” la cual se comenta poniendo un “#” delante de ella.

Después de ponerle la IP y la contraseña, el programa crea la clave RSA de conexión, genera el par publico/privado de claves. Si es la primera vez que se realiza la conexión, aparecerá un mensaje indicándonoslo, como se ve en la figura 1.17.

```
HERRAMIENTA DE ANÁLISIS Y GESTIÓN
Proyecto:      Proyectos/Manual

M E N U

1. Capturar trafico
2. Procesar trafico
3. Compartir claves
4. Preparar Experimento
5. Generar Experimento
6. Lanzar Experimento
7. Ejecutar comandos
8. Cambiar de Proyecto
9. Salir

Seleccionar número [1-9] 3
Dame la ip: 155.210.140.36
Dame usuario: test
Usando clave existente
Agregando información al fichero Known_hosts
# 155.210.140.36 SSH-2.0-OpenSSH_5.3
test@155.210.140.36's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'test@155.210.140.36'"
and check to make sure that only the key(s) you wanted were added.

Clave RSA pública copiada satisfactoriamente en 155.210.140.36
Could not open a connection to your authentication agent.
Pulsa una tecla para continuar █
```

Figura 1.17 Clave remota nueva

Las siguientes veces que se intente conectar, no hará falta introducir la contraseña.

```
Dame la ip: 155.210.140.36
Dame usuario: test
Usando clave existente
Clave RSA pública copiada satisfactoriamente en 155.210.140.36
Could not open a connection to your authentication agent.
Pulsa una tecla para continuar █
```

Figura 1.18 Clave remota conocida

Se tiene que asegurar que el equipo al que se quiere conectar tenga activado el servicio Ssh, si no es así se puede activar en el equipo con:

```
service sshd start (con permisos de root)
```

En caso de no tener el demonio anterior corriendo, puede salir el siguiente error de conexión:

```
/bin/ssh-copy-id: ERROR: ssh: connect to host 10.3.12.113 port 22: Connection refused
```

En este momento se tiene realizada la conexión segura a ese equipo, el mismo procedimiento tendría que realizarse para el resto de los equipos a los que después se quisiera modificar o controlar el tráfico de red, en el caso de que cada equipo tuviera un usuario y contraseña distinta, si todos los equipos remotos tuvieran el mismo nombre de cuenta de usuario cada uno con su propia contraseña distinta, se podría poner todas las IP's seguidas separadas por un espacio y de una sola vez nos realizaría todas las conexiones una detrás de otra.

7. Preparar experimento

En esta cuarta opción del menú: “Preparar experimento”, se permite tanto crear como editar los ficheros con extensión “.tabla” y solo deja editar los ficheros con extensión “.exp” (La creación de estos ficheros se hace en el punto siguiente).

Los ficheros “.tabla” serán los que contendrán los datos necesarios ordenados en columnas (separados por una coma) que permitirán generar los ficheros “.exp” (en la opción quinta del menú: “Generar experimento) que se lancen (en la opción sexta del menú: “Lanzar experimento) y que realizará las modificaciones del tráfico de red a equipos remotos. Si en “Nombre de tabla o experimento:” se pone “muestra.tabla” se podrá ver una cabecera indicativa de cada una de las columnas que se puede rellenar (imagen 19).

Si ya hemos generado experimentos, dentro de esta opción del menú los podemos editar y modificar igual que hacemos con las tablas.

```
HERRAMIENTA DE ANÁLISIS Y GESTIÓN
Proyecto:      Proyectos/Manual
Opción 4: Preparar experimento
-Tablas creadas:
-----
muestra.tabla
-Experimentos creados:
-----
experimento.exp muestra.exp
Para editar o crear una tabla, dame un nombre.tabla
Para editar un experimento, dame un nombre.exp
(Crear experimento Opción 5 del Menú)
Nombre de tabla o experimento: █
```

Figura 1.19 Preparar experimento

El programa abrirá el fichero que se le indique con un editor de texto dentro de la consola, “Nano” para que se pueda ver y hacer las modificaciones que se quieran sin tener que salir de la aplicación y así poder ir ajustando la tabla para que luego se genere el fichero del experimento que se ejecute.

Aquí se muestra una captura con editor Kate para ver los campos de la tabla más grandes y para una explicación mejor:

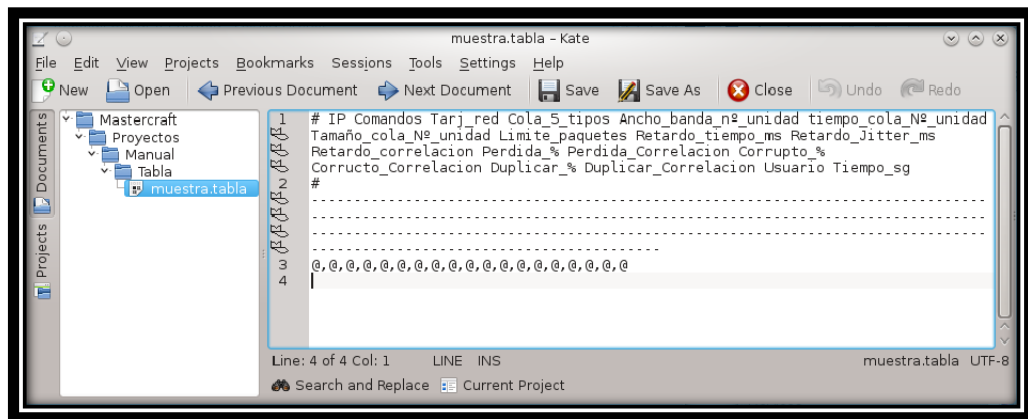


Figura 1.20 Muestra de Tabla

Se utilizan dos herramientas para manipular y controlar el tráfico de red: Traffic Control (**Tc**) y el Network Emulator (**Netem**).

Se genera una tabla con 19 columnas, a cada campo se le asigna el valor adecuado para configurar las herramientas **Tc** y **Netem**. A continuación se explica cada campo de la tabla:

Columna 1: (IP) Dirección IP a la cual se conecta con el objetivo de manipular y modificar el tráfico.

De la columna 2 a la 7 se especifican los campos necesarios para el Traffic Control (**Tc**). Se define la regla **Tc qdisc** (*queueing discipline*), con mecanismos de disciplinas de colas. Las colas determinan el orden en que se mandan los paquetes.

Columna 2: (Comandos) Comandos del **Tc**

- *add*: añade una política **qdisc**, **class** o **filter** a un nodo.
- *del*: elimina la política que se halla añadido con “add” y deja la que tenga la tarjeta de red por defecto.
- *change*: comparte la sintaxis de *add*, con la diferencia de que con esta opción el nodo no se puede mover.
- *remove*: una **qdisc** se puede eliminar especificando su *handle* o también conocido como *identificador*. Todas las subclases de una **qdisc** son eliminadas al igual que los filtros asociadas a ellas.
- *replace*: realiza una acción de *remove/add* a un nodo de manera atómica. Si el nodo no existe, se crea.
- *link*: esta opción solo está disponible para **qdisc** y realiza un reemplazo en el mismo sitio donde se encuentra el nodo.

Columna 3: (Tarj_red) Es la interfaz de red del ordenador al que se conecta, correspondiente al tráfico que se analiza y manipula (puede tener cualquier identificador).

Columna 4: (Cola_5_Tipos) Se especifica el tipo de cola que se utilizará. Pueden ser de cuatro tipos diferentes:

- *pfifo* o *bfifo*: son las más sencillas, un First In First Out, pueden estar limitadas en paquetes o en bytes.
- *red*: **Random Early Detection**, es un gestor de colas inteligente. Simula una congestión física donde se eliminan paquetes de manera aleatoria cuando se ocupa un ancho de banda cercano al ya configurado. Muy adecuado para aplicaciones con grandes requerimientos de ancho de banda.
- *sfq*: **Stochastic Fairness Queueing** reorganiza el tráfico que se encuentra en cola para que cada 'sesión' alcance a enviar un paquete en su turno. Funciona bien para un ancho de banda bastante elevado, consume pocos ciclos de CPU.
- *tbf*: **Token Bucket Filter**, es adecuado para frenar o retener el tráfico a una velocidad que se ha configurado. Adecuado para anchos de banda muy grandes.

Columna 5: (Ancho_banda_nº_unidad) Se corresponde al ancho de banda, se debe especificar su valor junto a la unidad de medida: kbps, mbps, kbit, mbit, bps. (El programa añade ese valor detrás del parámetro "rate").

Columna 6: (Tiempo_cola_Nº_unidad) Se corresponde al tiempo que se le asigne a la cola (se debe especificar la unidad de medidas: segundos (sec), milisegundos (msec) o microsegundos (usec)). (El programa añade ese valor detrás del parámetro "time").

Columna 7: (Tamaño_cola_Nº_unidad) Configura el tamaño de la cola en datos (de igual manera, se debe especificar la unidad de medida: kilobytes (kb), megabytes (mb), kilobits (kbit), megabits (mbit) o bytes (b)). (El programa añade ese valor detrás del parámetro "size").

Columna 8: (Limite_paquetes) (**n**) Se aplican las opciones seleccionadas a los **n** siguientes paquetes. (El programa añade ese valor detrás del parámetro "limit").

De la columna 9 a la 17 se especifican los campos necesarios para el Network Emulator (**Netem**). **Netem** permite realizar pruebas de tráfico mediante la emulación de distintas propiedades de la red, se configura para que todos los paquetes que salgan por una cierta interfaz puedan ser procesados.

Dispone de cuatro parámetros básicos:

1. Retardo (*delay*): añade retardo a cada paquete.
2. Pérdida (*loss*): elimina/desecha algunos paquetes.
3. Duplicación (*duplicate*): duplica algunos paquetes.
4. Corrupción (*corruption*): introduce error en un bit de manera aleatoria dentro de un paquete.

(El programa añade el comando “Netem” y sus parámetros obtenidos de la tabla).

Columna 9: (Retardo_tiempo_ms) Añade el retardo seleccionado a los paquetes que se generan por la interfaz de red seleccionada en la Columna 3. Se expresa en milisegundos (ms). (El programa añade ese valor detrás del parámetro “delay”).

Tiene parámetros opcionales que permiten introducir variaciones en el retardo y una correlación, ellos son:

Columna 10: (Retardo_Jitter_ms) Para especificar el jitter del retardo (expresado en milisegundos)

Columna 11: (Retardo_correlacion) La correlación del retardo (expresada en %). Se relaciona con el tipo de distribución que presenta el retardo (uniforme, normal, pareto, paretonormal, etc.).

El retardo en una red no tiene un comportamiento uniforme, por lo que es común utilizar distribuciones con el objetivo de describir la variación del retardo. Si no se especifica, la distribución por defecto que emplea **Netem** es la Normal.

Columna 12: (Perdida_%) En este campo se define la pérdida de paquetes en porcentaje (en %). Se añade una probabilidad de pérdidas (aleatoria) a los paquetes que salen por la interfaz de red seleccionada en la Columna 3. Se le puede añadir una correlación, pero actualmente esta opción no se utiliza debido a que presenta un mal funcionamiento. (El programa añade ese valor detrás del parámetro “loss”).

Columna 13: (Perdida Correlación) Se define la correlación de las pérdidas que se mencionó anteriormente. (El manual de **Netem** indica que esta opción está en desuso).

Columna 14: (Corrupto_%) Se definen los paquetes corruptos (en %). Permite la emulación de un ruido aleatorio introduciendo un error en cualquier posición en un porcentaje de paquetes seleccionados. En este caso también es posible añadir una correlación a través del propio parámetro, tal y como se especifica en la Columna 15. (El programa añade ese valor detrás del parámetro "corrupt").

Columna 15: (Corrupto_correlacion) Se define la correlación de los paquetes corruptos.

Columna 16: (Duplicar_%) Se define el porcentaje de paquetes duplicados (en %). Si se utiliza esta opción, el porcentaje de paquetes seleccionados se duplican antes de enviarlos a la cola. Como en los casos anteriores, es posible añadir una correlación a través del propio parámetro y esta se define en la siguiente columna. (El programa añade ese valor detrás del parámetro "duplicate").

Columna 17: (Duplicar_Correlacion) Se define la correlación de los paquetes duplicados.

Columna 18: (Usuario) Es el usuario con el que se vaya a ejecutar el "Tc" y "Netem" o es root o se le ha tenido que dar permisos con "visudo" en el fichero "sudoers" remoto. (Ver punto 6 de este manual, Compartir Claves).

Columna 19: (Tiempo_sg) En este campo se define el intervalo de tiempo en el que se suspende o bloquea la ejecución de un proceso, se emplea el comando *sleep* y se define en segundos. Este comando es muy utilizado cuando se requiere postergar el lanzamiento de algún comando o alguna línea de comandos. El intervalo de tiempo se indica por medio de un valor entero que se interpreta en segundos.

La tabla se tiene que rellenar en toda sus columnas, los campos que no se modifican o rellenan, se deben poner con el símbolo de @. Cuando la herramienta detecta en algún campo el símbolo @, salta al siguiente campo. Cada uno de los campos va separado por una coma ",".

En el caso de que el investigador necesite introducir una expresión compleja con más parámetros, o parámetros distintos que no estén contemplados en esta guía o en este programa, lo podría realizar rellorando las columnas 1, 2 y 3 como hemos indicado anteriormente, introduciendo su expresión compleja en la columna 4 y dejando el resto de columnas con el símbolo @ (Menos la 18 con el usuario y la 19 con el tiempo). El programa ejecutaría la tabla sin problemas y de esta manera se aumenta la potencia de la herramienta hasta donde se necesite.

Este es un ejemplo de cómo quedaría la tabla para que la interprete el programa y la pueda ejecutar.


```
GNU nano 2.3.4 Fichero: manual.tabla
155.210.164.158,add,eth0,bfifo,0,0,0,104kb,0,0,0,0,0,0,0,0,cliente1,10
155.210.164.158,del,eth0,0,0,0,0,0,0,0,0,0,0,0,cliente1,2
█
```

Figura 1.21 Ejemplo tabla

8. Generar experimento

En esta quinta opción del menú: “Generar experimento”, nos muestra un listado de las tablas que se tienen creadas para que se pueda elegir una de ellas escribiendo su nombre donde dice “Nombre de tabla (sin extensión) para generar experimento” se pulsa “enter” y el programa genera un fichero con el mismo nombre que la tabla y con extensión “.exp” que permita modificar el tráfico de red según los parámetros que se le hayan introducido en la tabla anteriormente indicada. Y que en la opción siguiente del menú nos permitirá ejecutar.

```
HERRAMIENTA DE ANÁLISIS Y GESTIÓN
Proyecto: Proyectos/Manual
Opcion 5: Generar Experimento
-Estas son las tablas para generar los experimentos.
Tablas creadas:
-----
manual muestra
Nombre de tabla (sin extensión) para generar experimento: manual
Experimento generado con el nombre manual.exp
Para ejecutarlo entra en la opcion '6. Lanzar Experimento'
Pulsa ENTER para continuar: █
```

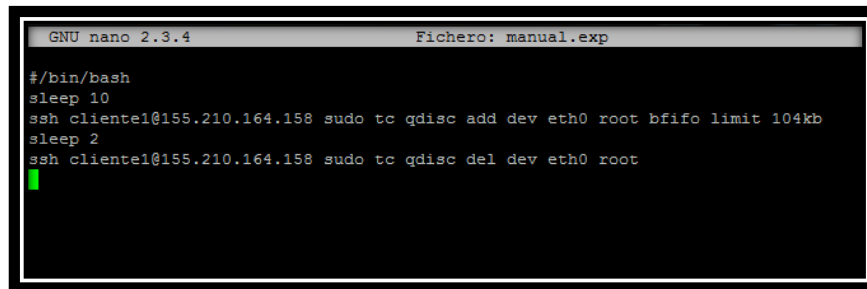
Figura 1.22 Generar experimento

La siguiente tabla:

```
GNU nano 2.3.4 Fichero: manual.tabla
155.210.164.158,add,eth0,bfifo,0,0,0,104kb,0,0,0,0,0,0,0,0,cliente1,10
155.210.164.158,del,eth0,0,0,0,0,0,0,0,0,0,0,0,cliente1,2
█
```

Figura 25.23 Tabla generada

Generaría el siguiente fichero experimento:



```
GNU nano 2.3.4 Fichero: manual.exp
#!/bin/bash
sleep 10
ssh cliente1@155.210.164.158 sudo tc qdisc add dev eth0 root bfifo limit 104kb
sleep 2
ssh cliente1@155.210.164.158 sudo tc qdisc del dev eth0 root
```

Figura 1.24 Editar experimento

Se puede ver, editar (opción anterior del menú) y comprobar que genera un programa que se puede ejecutar en la siguiente opción de menú o incluso desde la consola directamente, ejecutándolo con un “.” delante.

Se explica cómo se generan estos comandos a partir de las entradas de la tabla:

- a. Se esperan 10 segundos antes de lanzar la línea de comandos siguiente (**sleep**). (Última columna de cada línea).
- b. Se realiza la conexión remota segura por **Ssh** al equipo con dirección IP 155.210.164.158, utilizando el usuario con privilegios para ejecutar Ssh, Tc o Netem, como ya se ha explicado anteriormente, cliente1.
- c. Se emplea el comando **Tc qdisc**, pues el tráfico será modificado haciendo uso de políticas de cola.
- d. Luego se añaden las características que tendrá el tráfico, mediante el comando **add**.
- e. Con el comando **dev** se define la interfaz de red donde se harán las capturas, se corresponde con la aparece en la tabla de balanceo (eth0).
- f. Cuando se usan **qdisc** sin clases (classless) sólo pueden adjuntar los nodos que se crean en la raíz de un dispositivo, se emplea el termino **root**.
- g. La política de cola que se emplea es **bfifo**, un First In First Out en bytes.
- h. Como los campos siguientes están con @, pues lo siguiente sería limitar el tamaño de la cola (**limit**) a 104 kilobytes.
- i. Posteriormente se esperan 2 segundos para ejecutar la línea siguiente (**sleep**)
- j. se vuelve a conectar por **Ssh** al equipo con dirección IP 155.210.164.158 y en este caso, se elimina completamente (**del**) la política de cola de la interfaz eth0 que se definió en la línea anterior.

9. Lanzar experimento

En la sexta opción del menú: “Lanzar experimento” Dentro de este apartado 6. Balanceo de interfaz el último punto del menú del programa, se puede ejecutar la tabla preparada en el apartado anterior o cualquier tabla que se tenga ya generada, únicamente se tiene que asegurar que estén dentro de la carpeta de proyecto, dentro de la carpeta Tabla y que tenga la extensión txt.

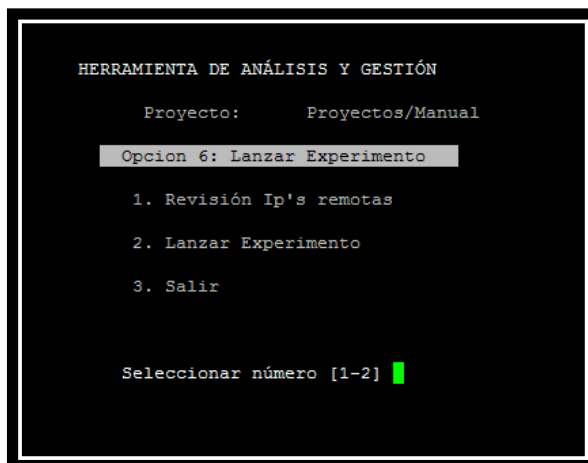


Figura 1.25 Menú: Lanzar experimento

La primera opción de este sub-menú, “Revisión Ip's remotas” es la misma que la tercera opción del menú principal: “Compartir claves”, (se puede ver cómo funciona en el punto 6 de este manual de usuario) únicamente se ha añadido aquí para que sea más fácil y rápido, sin tener que salir de este sub-menú y antes de lanzar el experimento, se puede revisar que las conexiones a las ips remotas funcionan bien y sin que nos pida contraseña.

La segunda opción del sub-menú, “Lanzar Experimento” muestra los experimentos generados y pide un nombre de experimento con su extensión para poder ejecutarlo. (Hasta que no termine de ejecutar el experimento no nos devolverá el control del ratón ni dejará hacer ninguna otra acción. Una vez que acabe su ejecución se podrá volver al menú principal).

```
HERRAMIENTA DE ANÁLISIS Y GESTIÓN
Proyecto:      Proyectos/Manual
Opcion 6: Lanzar Experimento
1. Revisión Ip's remotas
2. Lanzar Experimento
3. Salir

Seleccionar número [1-2] 2
-Estas son los experimentos generados:
Experimentos creados:
-----
experimento.exp manual.exp muestra.exp z.exp

Nombre del experimento (con .exp) a lanzar: █
```

Figura 1.26 Lanzar experimento

Cuando acabe el experimento da un mensaje de: Experimento “nombre”.exp ejecutado y pide que se pulse “ENTER” para volver al sub-menú que se estaba por si se quiere lanzar otro experimento.

10. Ejecutar comandos

En la séptima opción del menú: “Ejecutar comandos” se permite “sin tener que salir de nuestro programa” ejecutar programas u órdenes, las mismas que te permitiría ejecutar desde la consola de Linux. Se puede por ejemplo listar directorios, cambiar permisos, crear usuarios, lanzar generadores de tráfico, etc.

```
HERRAMIENTA DE ANÁLISIS Y GESTIÓN
Proyecto:      Proyectos/Manual
M E N U
1. Capturar trafico
2. Procesar trafico
3. Compartir claves
4. Preparar Experimento
5. Generar Experimento
6. Lanzar Experimento
7. Ejecutar comandos
8. Cambiar de Proyecto
9. Salir
Seleccionar número [1-9] 7
Elije opción
1) Lanzar comando, script o programa
2) Salir
Seleccionar número [1-2] 1
Introduce comando o script a ejecutar y pulsa enter: █
```

Figura 1.27 Ejecutar comando

Una vez ejecutado el comando, se vuelve a preguntar si se quiere ejecutar otro, sin salir de nuestro programa, ni de las opciones del menú.

```
Elije opción
1) Lanzar comando, script o programa
2) Salir
Seleccionar número [1-2] 1
Introduce comando o script a ejecutar y pulsa enter: █
```

Figura 1.28 Ejecutar comando 2

11. Cambiar proyecto

En la octava opción del menú: “Cambiar de proyecto” se permite entrar en otro proyecto distinto para ejecutar cualquier opción y cuando se sale de este proyecto se vuelve al que se está ejecutando en un primer momento.

```
HERRAMIENTA DE ANÁLISIS Y GESTIÓN
Proyecto:      Proyectos/Manual
M E N U
1. Capturar trafico
2. Procesar trafico
3. Compartir claves
4. Preparar Experimento
5. Generar Experimento
6. Lanzar Experimento
7. Ejecutar comandos
8. Cambiar de Proyecto
9. Salir
Seleccionar número [1-9] 8
Raiz: /home/david/Pruebas/Mastercraft
Nombre de proyecto: p1
```

Figura 1.29 Cambiar de proyecto

En la parte superior del menú principal se puede ver siempre en que proyecto se está, una vez se acabó con lo que se tenga que hacer en el proyecto “p1”, pulsando 9 Salir, se volvería al menú del proyecto “Manual”.

```
HERRAMIENTA DE ANÁLISIS Y GESTIÓN
Proyecto:      Proyectos/p1
M E N U
1. Capturar trafico
```

Figura 1.30 proyecto

Si no es esto lo que se quiere hacer, sino que se quiere cerrar un proyecto y entrar en otro, se sale del programa y se vuelve a entrar, poniendo el nombre del otro proyecto.

La última opción novena del menú principal “Salir”, permite salir del programa.

Anexo 3 Tablas y Figuras

Primero se muestran las 6 tablas con la información de las capturas de hitos, que aparecen en los experimentos del tema 3. Las tres primeras son envío de paquetes del cliente al servidor y las tres siguientes, del servidor al cliente. Tenemos tamaño de paquetes de los envíos y número de paquetes en cada una de las acciones.

Los distintos hitos que se capturaron son los siguientes: Estar quieto, correr, caminar, volar, dar espadaos, soltar flechas, cavar una fila, rellenar una fila, comer, morir ahogado, morir en lava, morir cayendo desde lo alto, nadar, cortar un árbol, que se quemen zombis, matar zombis tirándoles flechas, hacer login logout, perder vida, hablar por el chat del juego.

Tamaño paquete	Acciones					
	Quieto	Correr	Caminar	Volar	Espadazos	Flechas
43					186	
44	1149	1			1124	1028
46		2				
48	30	29	29	30	30	30
52	5				26	134
53						28
61						27
68	58	1203	1183	1213	58	54
76						4

Tamaño paquete	Acciones					
	Cavar fila	Rellenar fila	Comer	M. ahogado	M. lava	M. altura
40				40	3	2
43	1399	54				
44	888	721	315	586	29	47
46					2	
48	34	30	8	21	4	4
52	54	19	31		1	
53	180		3			
61		54	34			
68	455	465	16	52	57	42
76	8		1			11

Tamaño paquete	Acciones						
	Nadar	Cortar arbol	Quemar zombie	Flechas zombie	Login	Perder vida	Chat
40					20		
42					3		
43		1157		9			
44		923	397	1645	1420	506	1169
46				17			
48	32	34	10	54	37	16	31
50					3		
52		281		293	50	32	
53		104		33	1		
55					1		
58				16			
60					1		
61				32	4		
68	1149	97	20	172	71	86	57
76	151	92		82	4	76	15

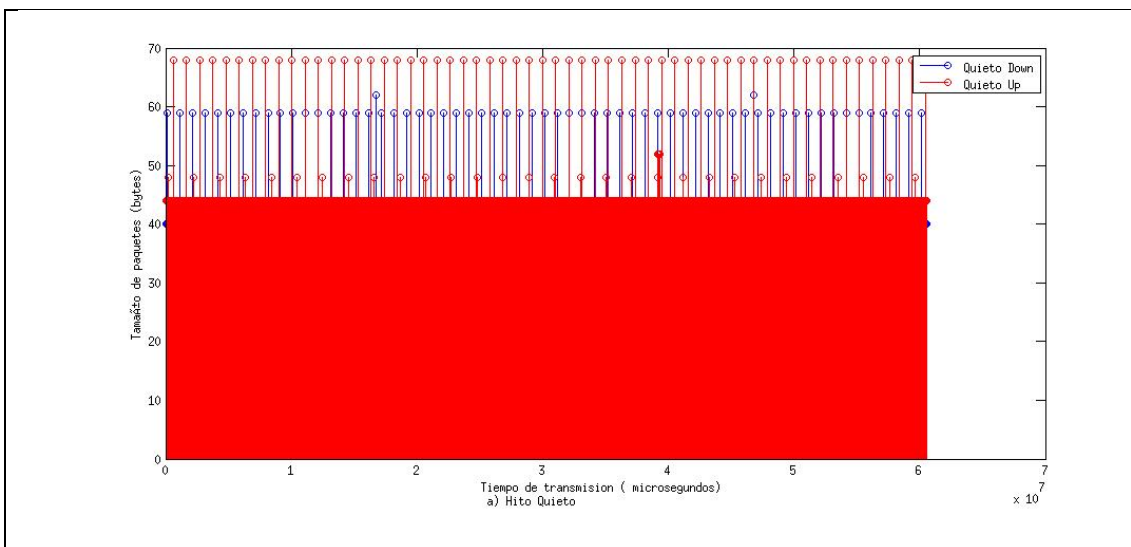
Tamaño paquete	Acciones					
	Quieto	Correr	Caminar	Volar	Espadazos	Flechas
40	1240	1130	1137	1068	1228	1139
45						5
47						27
48	30	29	29	27	30	352
52		4				
55						3
56						8
59	60	58	57	54	60	60
62		2		2	2	2
64						2
66						8
68		28	24	51		
77 a 1000		164	133	207		87

Tamaño paquete	Acciones					
	Cavar fila	Rellenar fila	Comer	M. ahogado	M. lava	M. altura
40	1463	1149	341	72	76	102
45	7					
46	2					
47			3		2	
48	344	28	12	489	12	4
49	68					
50	19		1	11		
51	14					
52	46	38	21		2	1
53	2					
56	33			12		
57			3			
58	5	1		1		
59	87	58	16	14	7	7
60	12					
61	2					
62	3	2		1		1
64	2		4			
65	2			7		
67				18		
68	5	4				
69	38					
70	18					
71	7	54				
72					3	
73				5		
77 a 1000	89	62	24	13	4	2

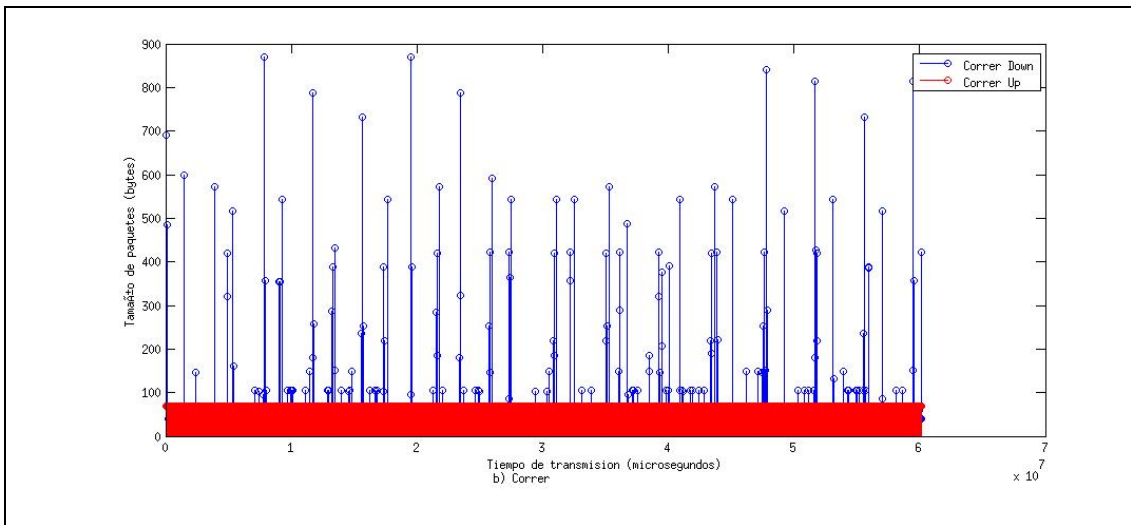
Tamaño paquete	Acciones		
	Nadar	Cortar arbol	Chat
40	1104	1448	1270
45		22	
46		1	
48	32	38	30
50		6	
52		105	
53	1	9	
57		22	
58		1	
59	66	74	62
62	2	1	2
67		1	
68	8		
69		32	
71		23	
75		1	
76		2	
77 a 1000	227	31	15

A continuación tenemos las gráficas generadas por Matlab de los 20 hitos

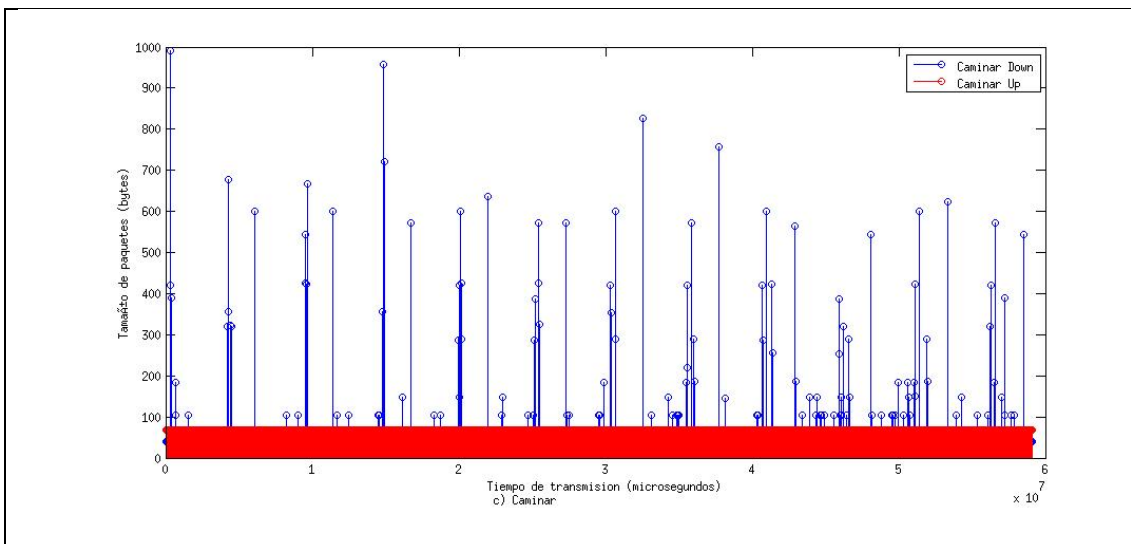
1 Quieto



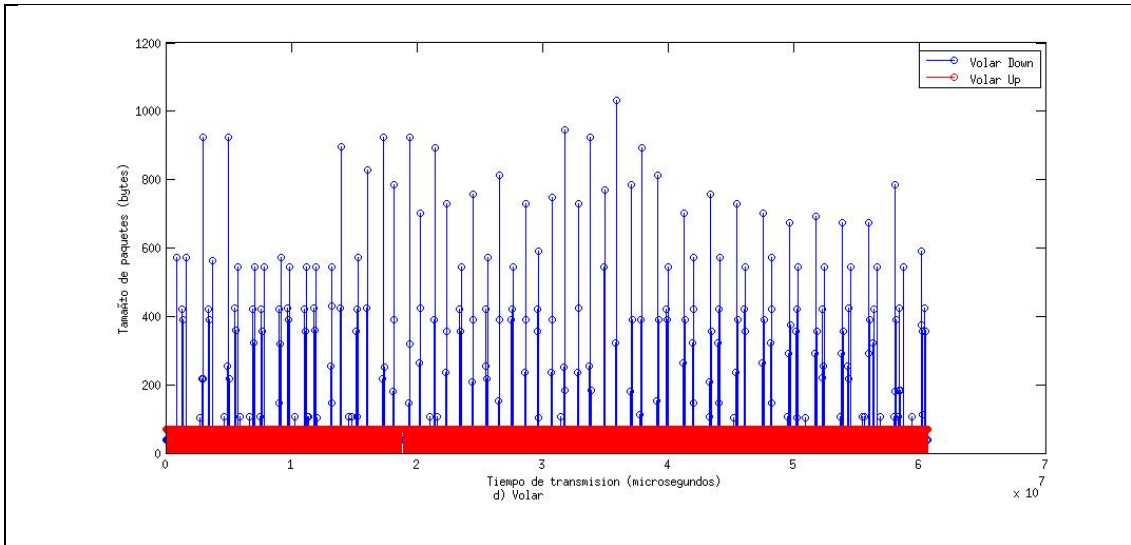
2 Correr



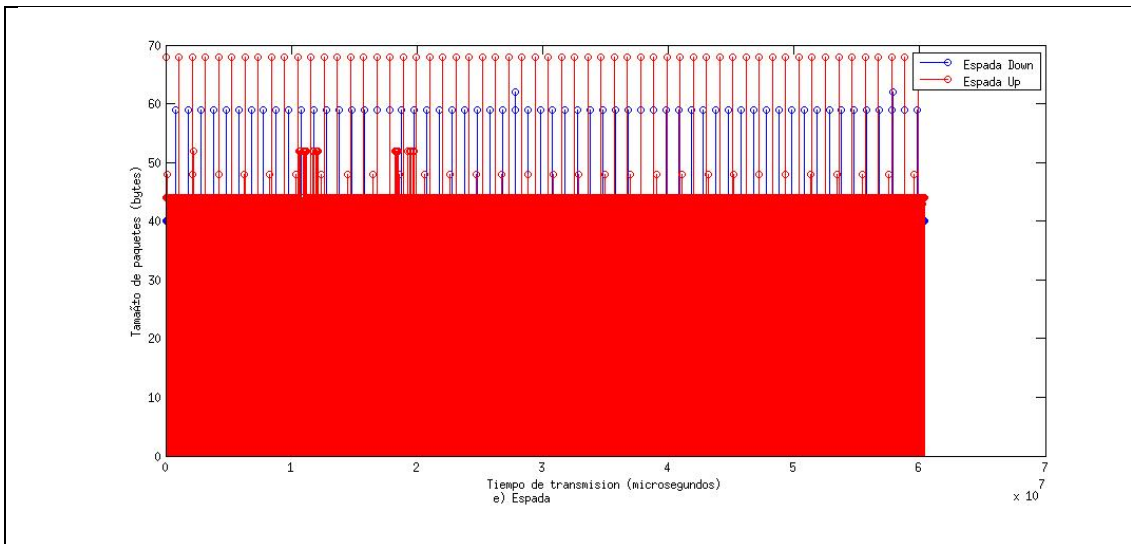
3 Caminar



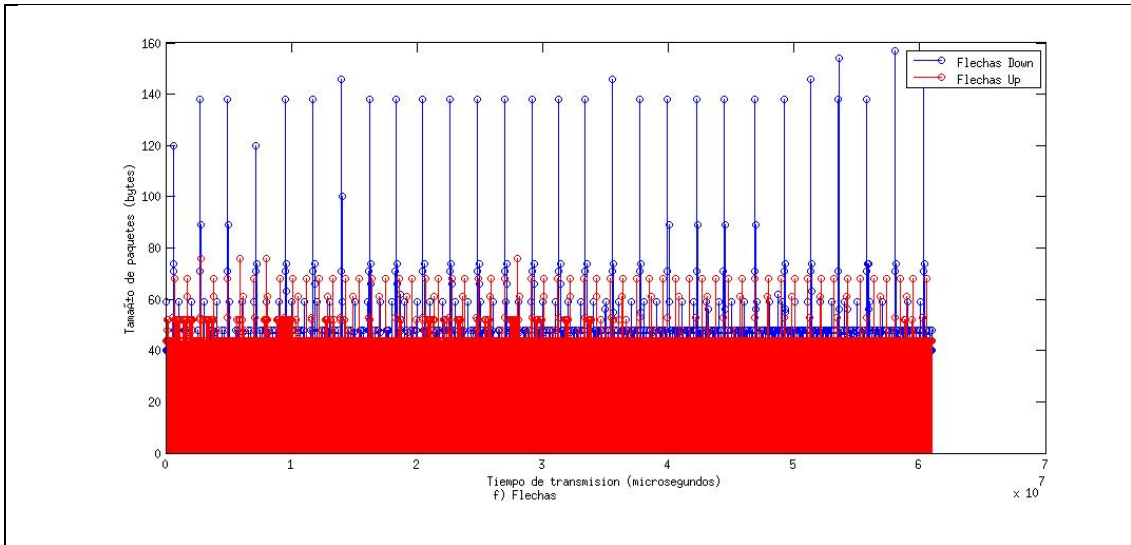
4 Volar



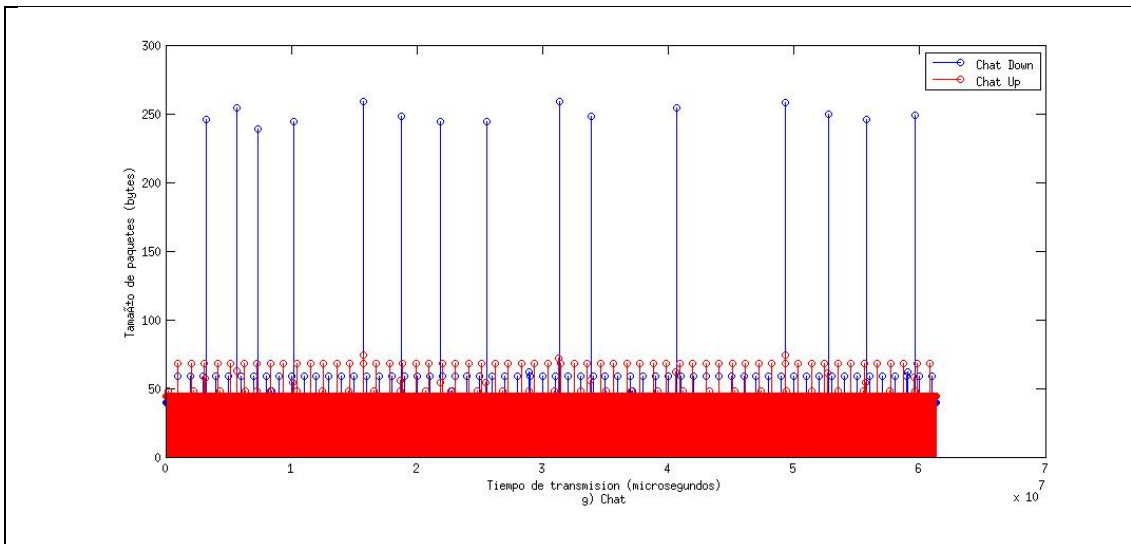
05 Dar espada



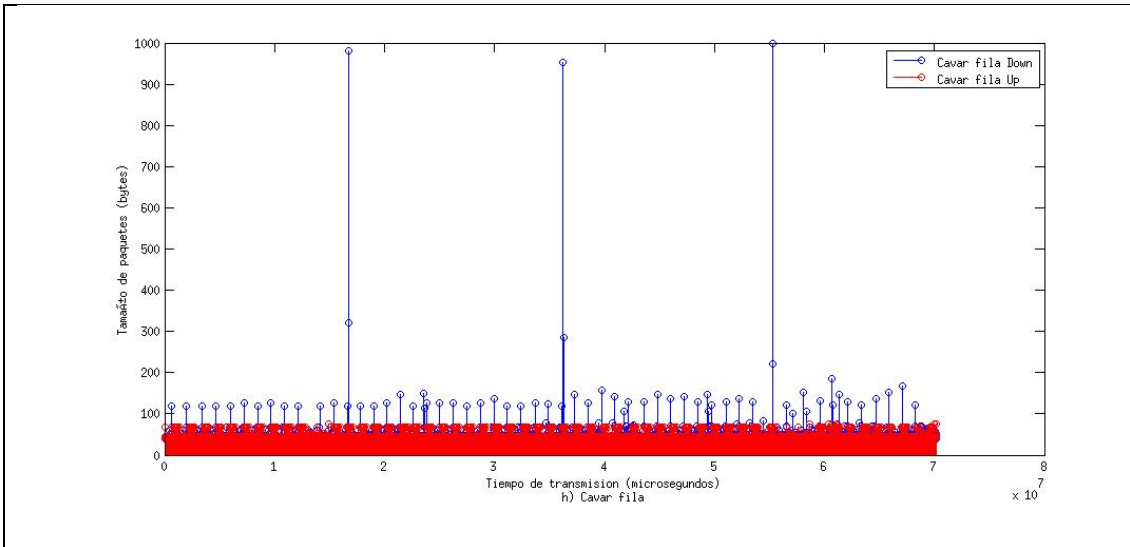
06 Lanzar flechas



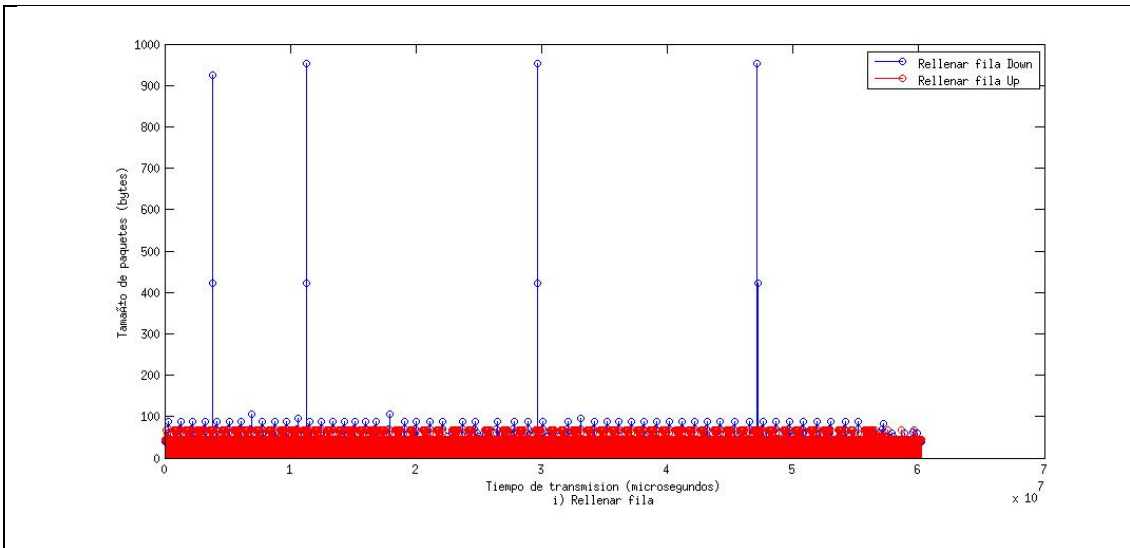
07 Hablar chat



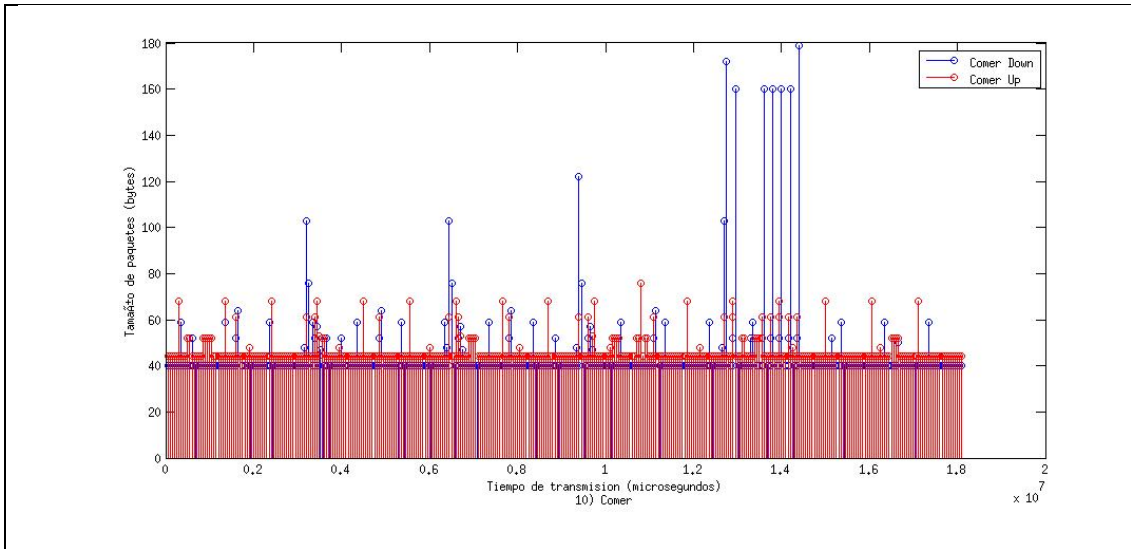
08 Cavar una fila de tierra



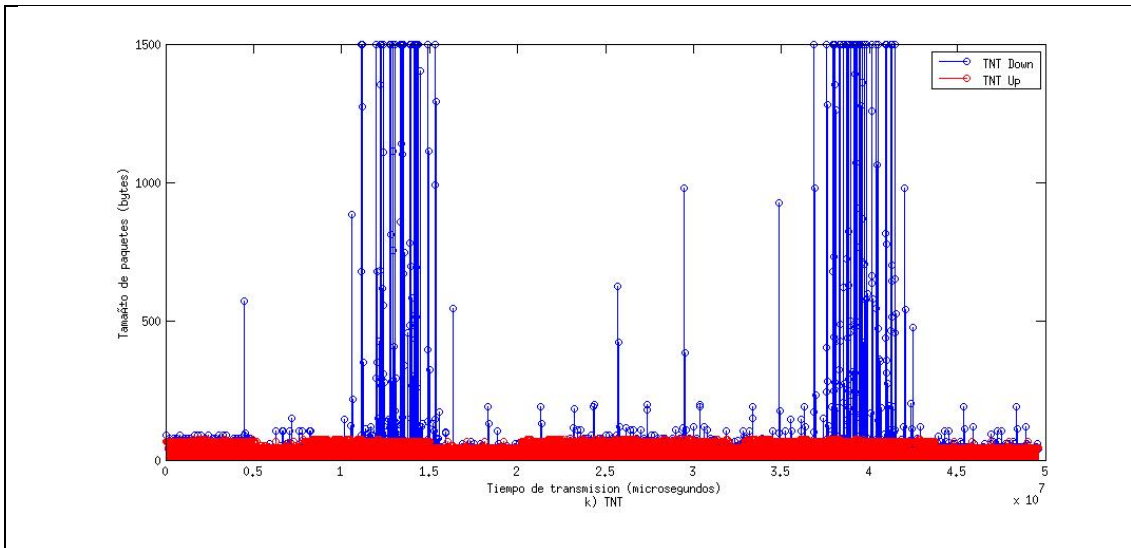
09 Rellenar una fila de tierra



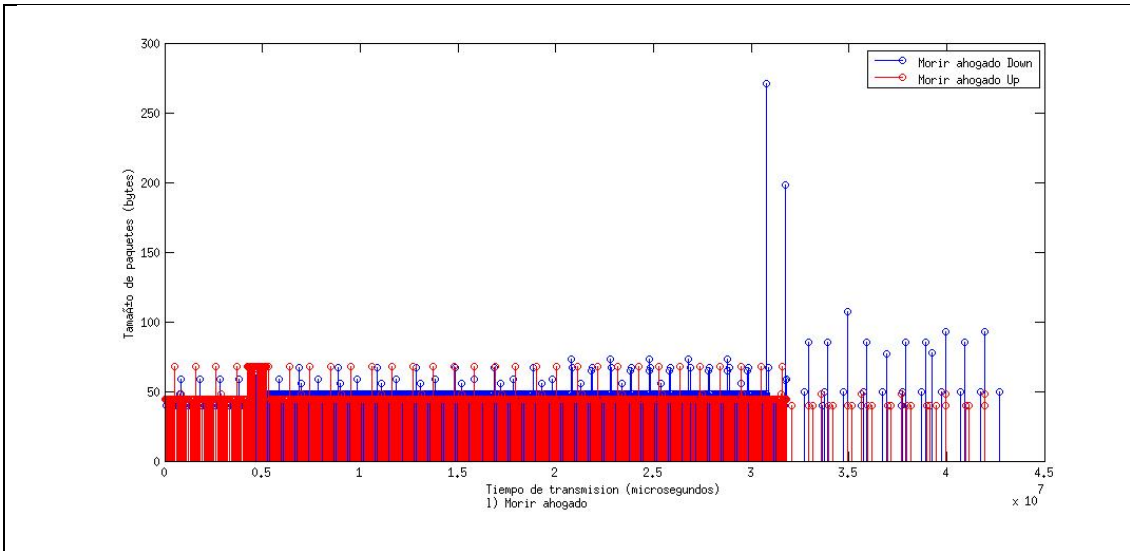
10 Comer



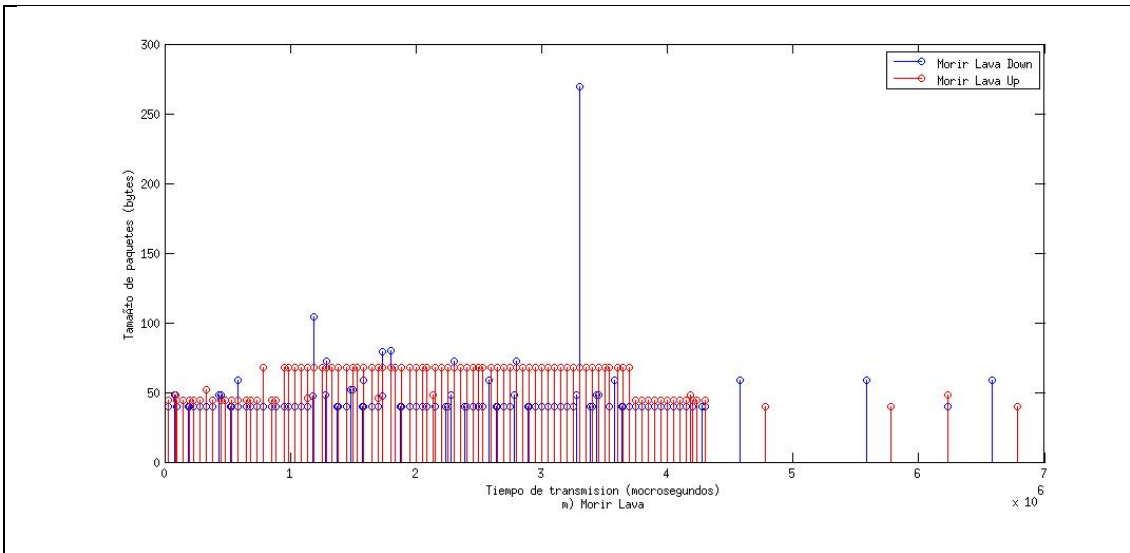
11 Tnt, explosiones



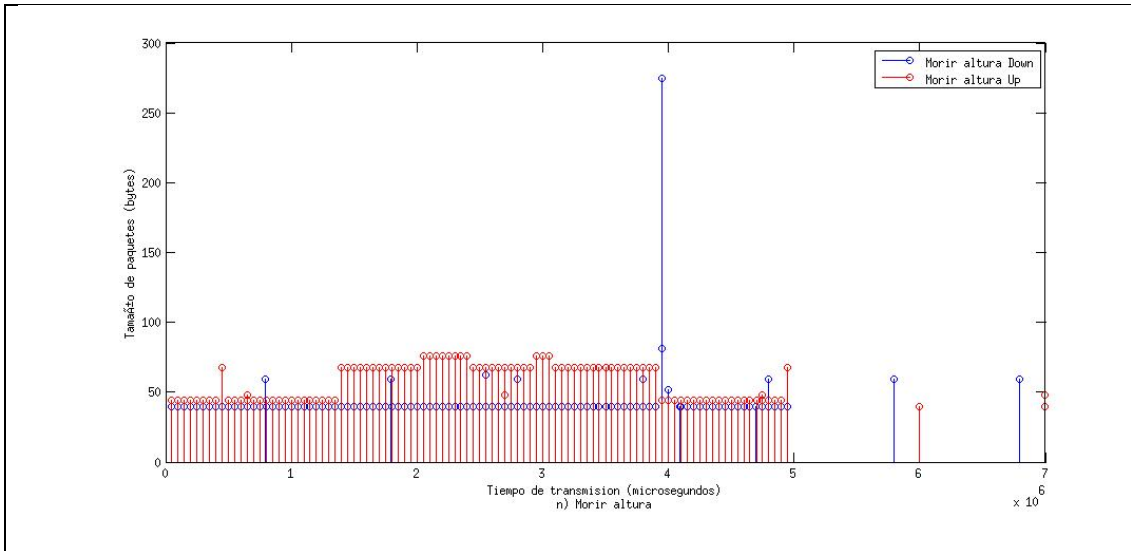
12 Morir ahogado



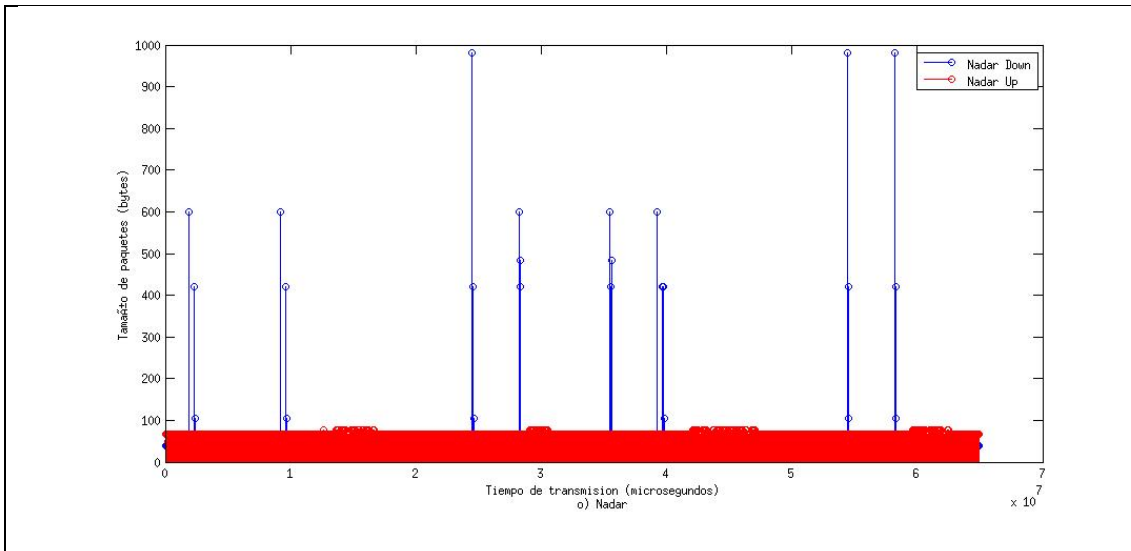
13 Morir cayendo en lava



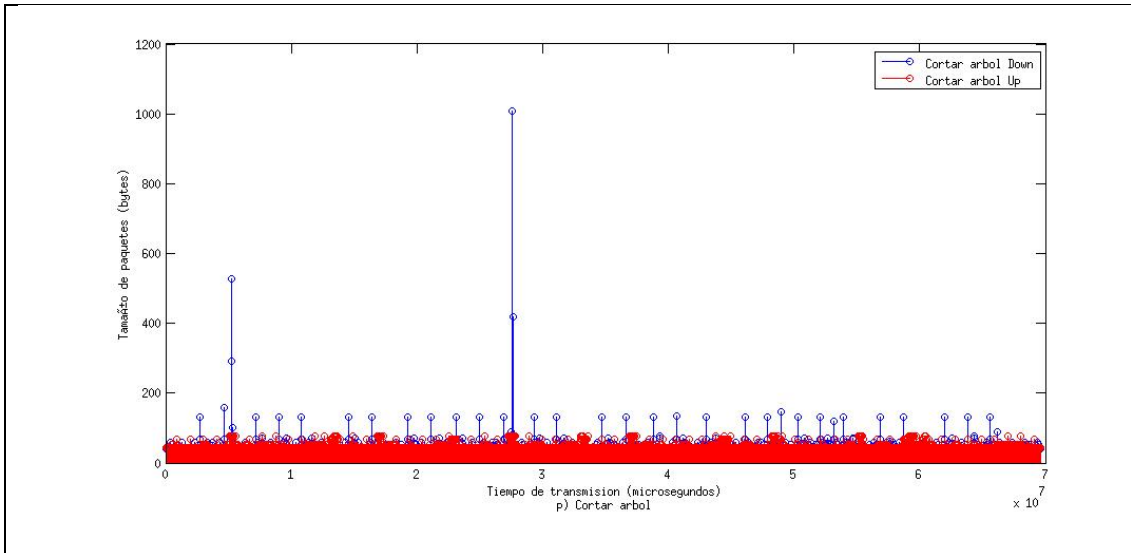
14 Morir cayendo de altura



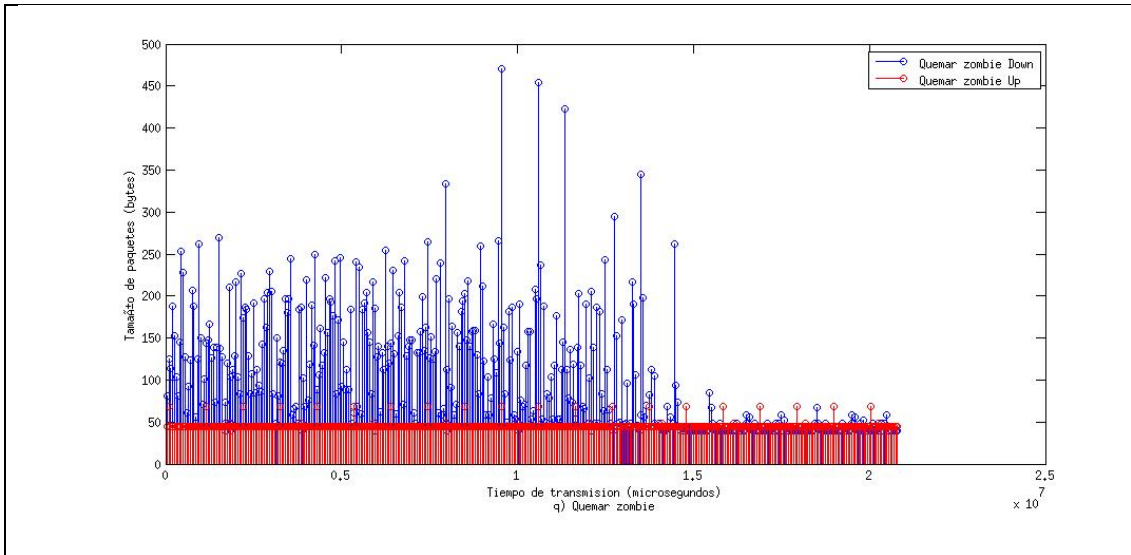
15 Nadar



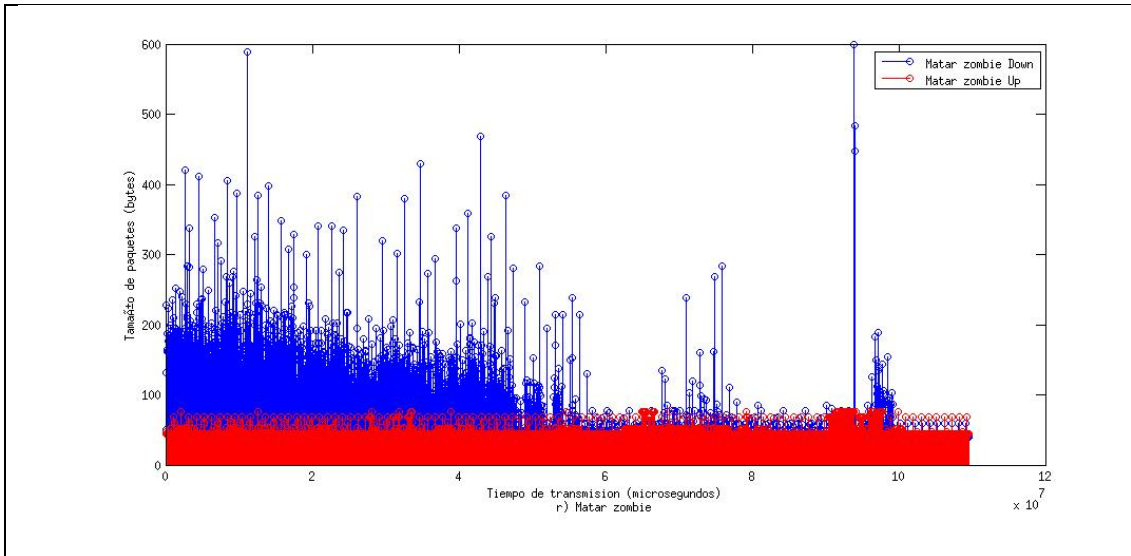
16 Cortar arboles



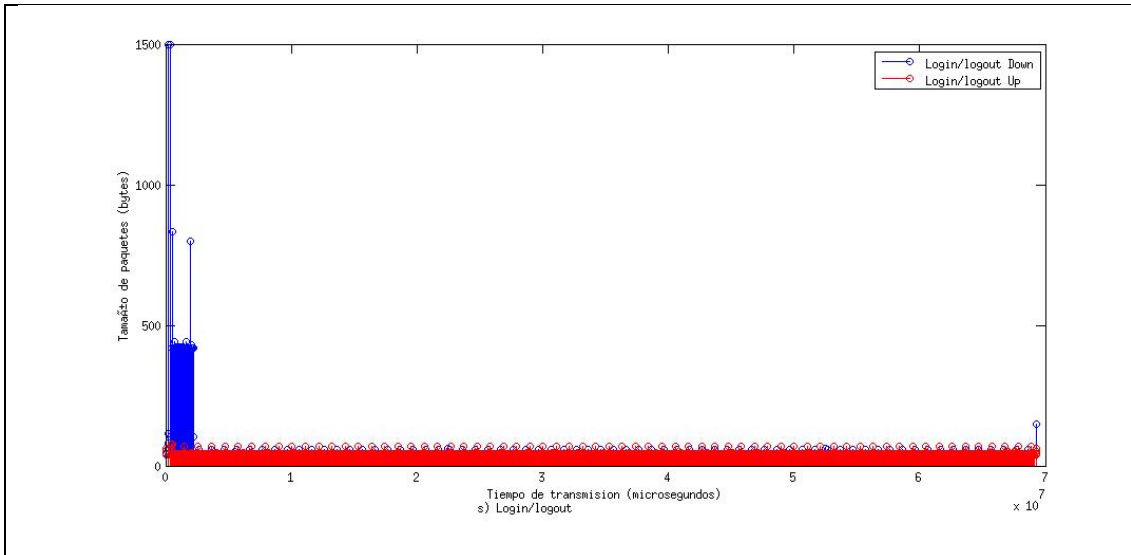
17 Quemarse Zombies



18 Matar Zombies con flechas



19 Login/logout



20 Vida

