Máster en Tecnologías de la Información y Comunicaciones en Redes Móviles

Programa Oficial de Posgrado en Ingeniería de Telecomunicación

---

METODOLOGÍA PARA EL MODELADO Y EL ANÁLISIS DE FLUJOS IP MULTIMEDIA: MEDIDAS DE CALIDAD

Luis E. Sequeira Villarreal

Director:
Julian Fernández Navajas
Ponente:
José Ruiz Más

Departamento de Ingeniería Electrónica y Comunicaciones
Universidad de Zaragoza
Noviembre 2011

Dedicatoria

A María Luisa

Agradecimientos

Por su apoyo y financiamiento para el desarrollo de este trabajo:

Fundación Carolina
Universidad de Zaragoza
Universidad Latina de Costa Rica

Resumen

"Metodología para el modelado y el análisis de flujos IP multimedia: medidas de calidad"

El presente trabajo propone una metodología para el modelado y el análisis de flujos IP multimedia que se transportan mediante tecnologías WiFi. Dicha metodología tiene como objetivo la obtención de modelos de tráfico multimedia de manera experimental y generación de tráfico en entornos controlados de laboratorio para una aproximada reconstrucción de los escenarios reales. Además, se propone la adaptación de todo el proceso a otros tipos de tráfico y la automatización de los mismos.

En primer lugar, para la realización del trabajo se describen algunas de las herramientas más utilizadas, y otras más recientes, que permiten la captura de datos, la gestión y generación de tráfico, así como el procesado de la información. También se describen una serie de aplicaciones reales escogidas en el presente trabajo, que generan los flujos IP con requerimientos temporales entre los extremos de la comunicación (voz, vídeo y *streaming*).

Por otro lado, dicha metodología se pone a prueba para la construcción de modelos de flujos IP de voz y vídeo, obteniendo resultados que permiten generar el mismo tráfico sin la necesidad de la aplicación. También, se utiliza para modelar las características del escenario. Como ejemplo se realiza el modelado de un *buffer* en un punto de acceso WiFi. En él se determinan algunas de sus características técnicas y funcionales más relevantes.

Con dichos flujos se realizan medidas de calidad en cuanto a pérdidas en entornos 802.11, en diferentes escenarios inhalámbricos reales. Estas medidas se comparan con los valores teóricos del protocolo de acceso al medio utilizado por dicho estándar CSMA/CA (*Carrier Sense Multiple Access Collision Avoidance*) para analizar el comportamiento real y encontrar las posibles diferencias.

Al final se plantean las conclusiones obtenidas y algunas de las líneas futuras de investigación que se relacionan con los fenómenos observados.

Índice de contenidos

Índice de figuras

Índice de tablas

1 Introducción

1.1 Descripción del problema

La creciente demanda de acceso a aplicaciones o servicios en entornos de red, según mencionan [3], [4] y [5], genera que las empresas y los centros de investigación tengan la necesidad de verificar la calidad de las comunicaciones IP con la finalidad de poder tomar decisiones que permitan implementar nuevos servicios o ampliar los ya existentes.

Por otro lado, los servicios que actualmente se brindan a través de Internet o en un entorno corporativo como lo son los sistemas de videoconferencia, voz sobre IP o servicios de *streaming* [4] están asociados a una serie de protocolos que definen aspectos como el transporte, el tipo de *codec* que se utiliza o los niveles de compresión de video o voz.

El conjunto de todas estas técnicas que permiten brindar dichos servicios, tiene un efecto importante en la manera en que la información se propaga por la red. Por esto, es necesario conocer el comportamiento del tráfico en la red, para poder hacer mediciones de calidad en cualquier enlace sin generar un congestionamiento que degrade los servicios ya existentes, o bien, poder valorar la incorporación de un nuevo servicio a la red.

De tal manera, es necesario estudiar y seleccionar un conjunto de casos de uso que resulten significativos para la utilización de flujos IP multimedia principalmente en aquellos que tengan requerimientos temporales entre los extremos de la comunicación, ya que este tipo de servicios tienen un crecimiento importante en cuanto a la demanda por los usuarios.

Las universidades y departamentos de I+D tienen un interés especial en poder modelar el tráfico procedente de ciertas aplicaciones [6], ya que obteniendo los modelos correspondientes, estos se pueden reproducir, y con esto, ser puestos a prueba en diferentes escenarios que permitan el análisis del comportamiento del tráfico [7] y [8]. Además, en el desarrollo de nuevos protocolos y técnicas de transmisión de datos es necesario realizar una gran cantidad de pruebas con diferentes tipos de tráfico.

Para que un protocolo o técnica de cualquier índole se incorpore de forma operativa y permanente en una red, debe ser minuciosamente optimizado y depurado. La mejor manera de lograr esto, es realizar todas las pruebas en un entorno real, ya que de este modo se valoraría todos lo efectos introducidos por los equipos, los sistemas operativos, aplicaciones y los protocolos utilizados [9]. Sin embargo, las pruebas en entornos reales dejan de ser factibles cuando la cantidad de nodos crece a niveles comparables con un entorno típico de operación de ciertas aplicaciones y los costes se disparan notablemente.

Otra opción que comúnmente se utiliza en la simulación de entornos de redes, en este ámbito se encuentran diversas aplicaciones que facilitan esta labor, este es el caso de OPNET [10], NS-2 [11], Glomosim, entre otros. Los simuladores presentan la ventaja que facilitan la realización de las pruebas, la escalabilidad del sistema, además de garantizar de manera ágil la repetibilidad de las mismas, por otro lado, reducen enormemente los costes. Sin embargo, presentan la desventaja del alto consumo de recursos computacionales y que no toman en cuenta los sistemas operativos ni las aplicaciones que se desean analizar.

Otra de las técnicas a las que los investigadores recurren es a la emulación de entornos de red [12], [13] y [14]. En este caso se hace uso de máquinas virtuales con sistemas operativos completos y las aplicaciones que se deseen, por lo que el tráfico presenta los efectos del sistema operativo y las aplicaciones utilizadas.

Queda claro que las medidas del tráfico, el desarrollo de protocolos y el análisis de calidad en entornos de red es un contexto que manejan los centros de investigación de diferente manera, sin embargo, los procedimientos empleados para el modelado y análisis en los deferentes entornos de pruebas tienen mucho en común, a pesar que la técnica utilizada sea diferente.

Es por este motivo que surge la necesidad de una serie de procedimientos, que contemplados en una metodología, brinden los aspectos necesarios para el modelado de diferentes tipos de tráfico, que permita el análisis transparente de diversas aplicaciones y que sea fácilmente adaptable a cada una de las técnicas utilizadas por los investigadores. Por otro lado, que facilite, de manera sistemática, el análisis de la calidad de los enlaces de red

y el impacto de la variación del tráfico en ellos.

1.2 Objetivos

1. Obtener los procedimientos necesarios para la realización de medidas en diferentes escenarios de red, que permitan modelar sus tecnologías y las aplicaciones utilizadas.

2. Realizar medidas de tráfico de flujos IP en diferentes escenarios de red comúnmente utilizados por empresas y grupos de investigación con la finalidad de determinar su calidad.

3. Proponer una metodología para modelar y analizar la calidad de flujos IP con restricciones temporales en distintos escenarios de red.

1.3 Alcances y condiciones de trabajo

1.3.1 Alcances

El presente trabajo tiene como finalidad brindar una metodología que permita el análisis y modelado de flujos IP. Dicha metodología facilitará a los investigadores, la obtención de medidas en entornos de red, además, se adapta a diversas técnicas y tecnologías comúnmente utilizadas por departamentos de investigación.

1.3.2 Condiciones de trabajo

Las pruebas realizadas a lo largo de este trabajo se realizan con el objetivo de demostrar la veracidad de la propuesta planteada, además, han ayudado a depurar y corregir dicha propuesta.

El desarrollo de este trabajo se realiza en el Grupo de Tecnologías de las Comunicaciones (GTC) de la Universidad de Zaragoza, por lo tanto, la infraestructura, los equipos y las herramientas utilizadas se limitan a la disponibilidad de las mismas, por parte de dicho grupo de investigadores.

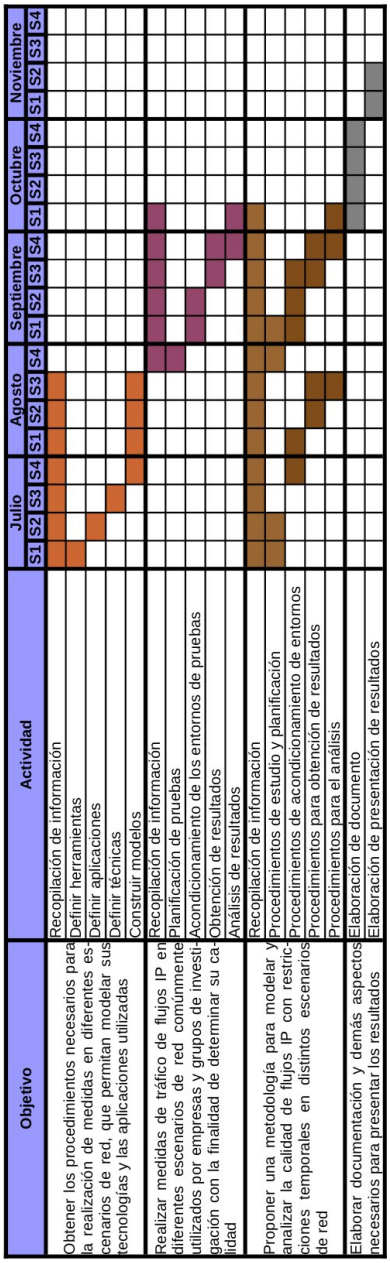
1.4 Estructura de la memoria

El presente trabajo consta de siete secciones, la primera describe el problema de estudio, plantea los objetivos del trabajo, los posibles alcances y el entorno en el cual se desarrolla. La segunda sección describe las principales herramientas utilizadas para el análisis y la implementación de las pruebas realizadas.

Posteriormente, en la tercera sección, se propone una metodología para el análisis y el modelado de flujos IP multimedia en tiempo real, de la cual se extrae un procedimiento que se puede consultar en el apéndice B. La sección cuatro y cinco muestra la obtención de los modelos de VoIP, vídeo de televigilancia, *streaming* y de *buffer* utilizando la metodología planteada.

La sección seis presenta una serie de medidas que se pueden realizar en diferentes escenarios de red para el análisis de la calidad utilizando los modelos anteriormente obtenidos. En la sección siete se presentan las conclusiones y las futuras líneas de investigación.

1.5 Cronograma de actividades



Objetivo	Actividad	Julio				Agosto				Septiembre				Octubre				Noviembre			
		S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
Obtener los procedimientos necesarios para la realización de medidas en diferentes escenarios de red, que permitan modelar sus tecnologías y las aplicaciones utilizadas	Recopilación de información																				
	Definir herramientas																				
	Definir aplicaciones																				
	Definir técnicas																				
	Construir modelos																				
Realizar medidas de tráfico de flujos IP en diferentes escenarios de red comúnmente utilizados por empresas y grupos de investigación con la finalidad de determinar su calidad	Recopilación de información																				
	Planificación de pruebas																				
	Acondicionamiento de los entornos de pruebas																				
	Obtención de resultados																				
	Análisis de resultados																				
Proponer una metodología para modelar y analizar la calidad de flujos IP con restricciones temporales en distintos escenarios de red	Recopilación de información																				
	Procedimientos de estudio y planificación																				
	Procedimientos de acondicionamiento de entornos																				
	Procedimientos para obtención de resultados																				
	Procedimientos para el análisis																				
Elaborar documentación y demás aspectos necesarios para presentar los resultados	Elaboración de documento																				
	Elaboración de presentación de resultados																				

Tabla 1: Cronograma de actividades.

2 Herramientas para la implementación y el análisis

Aspectos como la repetibilidad y la exactitud están estrechamente ligados a los entornos de pruebas y de análisis científico. Los investigadores tienen la necesidad de reproducir de manera flexible y escalable, diversos tipos de pruebas para diferentes condiciones en entornos controlados que permitan el análisis de los fenómenos estudiados, como mencionan algunos autores [9] y [12].

Por esto es necesario escoger un conjunto de herramientas que permitan facilitar la obtención de resultados para su posterior análisis. Dichas herramientas deben de ser seleccionadas con mucho cuidado ya que de ellas dependerá en cierta medida la exactitud de los datos obtenidos.

2.1 Herramientas de captura de datos

2.1.1 TCPDUM

TCPDUMP [15] es una herramienta para captura del tráfico que circula por la red en tiempo real (de los paquetes transmitidos y recibidos en una determinada interfaz de red). Dicha herramienta carece de interfaz gráfica, esto lo convierte en una de las aplicaciones preferidas cuando se quiere usar los mínimos recursos posibles [6], además, de ser idónea para la captura de paquetes en forma desatendida, ya que se gestione por línea de comandos.

Esta aplicación se encuentra disponible para casi todos los sistemas operativos (en Windows se llama *WinDUMP*), hace uso de la librería *libpcap* en el casos se sistemas UNIX y *winpcap* para Windows, además, es la encargada de las capturas de paquetes. Esta herramienta permite la depuración de la salida obtenida por medio de filtros, permitiendo filtrar capturas de puerto específico, o filtrando por tipo de protocolo, dirección origen o destino, en una interfaz en específico y otros.

2.1.2 Wireshark

Wireshark [16] es un programa de análisis, mantenido bajo licencia GNU GPL (*GNU General Public License*), también hace uso de las mismas

librerías de captura de paquete de las que se utilizan en TCPDUMP, dependiendo del sistema operativo. A diferencia de TCPDUMP, Wireshark permite su gestión mediante una interfaz gráfica amigable con el usuario sin la posibilidad de una gestión desatendida, además, posee funciones de filtrado y análisis de tráfico. Es compatible con el formato de archivo que utiliza TCPDUMP y reconoce una gran cantidad de protocolos.

Otra característica interesante para los investigadores es que permite la exportación de los archivos de capturas a diferentes formatos de aplicaciones orientadas al análisis matemático o de bases de datos, algo que puede resultar útil para realizar un análisis más detallado, como por ejemplo, cálculos de retardos, MOS, estadísticas y otras magnitudes que se puedan extraer de la captura de paquetes en la red.

2.2 Gestión y generación de tráfico

2.2.1 TC y NETEM

TC permite la gestión del tráfico en una interfaz determinada [5], [14], [9] y [12], limita el tráfico a un ancho de banda determinado haciendo una gestión de colas, clases y filtros de manera que se controle la forma con la se envía o reciben datos.

NETEM [17] *Network Emulator* es un emulador de red que está contenido en el kernel de linux a partir de la versión 2.6.7, por lo que todas las versiones actuales de linux poseen dicha aplicación integrada. NETEM es una extención de TC (*Traffic Control*), la herramienta de control de tráfico de linux, la cual está contenida en el paquete *iproute2*. NETEM permite introducir retardos y pérdidas sobre la transmisión de paquetes en una interfaz determinada [12].

2.2.2 JTG

El JTG (*Jugi's Traffic Generator*) es un generador de tráfico IP gestionado por línea de comandos, desarrollado por la Universidad de Helsinki [18], permite generar tráfico paramétrico y reproducir tráfico a partir de archivos con intervalos de tiempo y tamaños de paquetes a enviar en cada intervalo de tiempo [14]. Por otro lado, provee un receptor para recoger el tráfico, a

partir del cual se pueden calcular estadísticas. El JTG incluye un programa llamado *jtg_calc* que a partir del *log* de paquetes recibidos realiza un análisis estadístico de pérdidas, retardos y jitter.

2.2.3 D-ITG

Otro generador de tráfico IP multiprotocolo, desarrollado por la Universidad de Nápoles, es el D-ITG [19]. Este generador también permite el análisis de los datos enviados o recibidos, así como realizar capturas en ambos extremos de la comunicación, además, permite realizar medidas en un solo sentido OWD (*one-way-delay*) y de ida y vuelta RTT (*round-trip-time*) [12] y [13]. Puede ser gestionado mediante línea de comandos o por medio de GUI (*Graphical User Interface*) que puede ser obtenida en la misma página web del proyecto.

2.2.4 ETG

Recientemente, en el GTC (Grupo de Tecnologías de la Comunicación) de la Universidad de Zaragoza, grupo en el cual se desarrolla el presente trabajo, se ha implementado una herramienta denominada ETG (*E2E Trafic Generator*) [20], inicialmente implementada para el análisis de enlaces en el transporte de voz sobre IP y que actualmente se ha generalizado para generar los de flujos IP del trabajo.

Esta herramienta permite calcular parámetros objetivos (retardo, jitter, y tasa de pérdidas) y subjetivos de QoS (factor R, MOS). Se trata de una herramienta capaz de realizar comunicaciones E2E entre usuarios mediante el envío y recepción de varias secuencias de ráfagas de tráfico UDP. Permite generar tráfico OWD y RTT, así como capturas en ambos puntos de la comunicación, además, mediante el análisis del tráfico recibido se obtienen los parámetros de retardo, pérdidas y jitter, con los cuales podemos calcular el factor R que determina el MOS para cada comunicación.

Una ventaja de esta aplicación es la facilidad que presenta para la repetibilidad de la pruebas ya que contempla mecanismos que permiten la automatización de ciertas funciones facilitando el trabajo a los investigadores, como por ejemplo: cada flujo se podrá reiterar cada cierto tiempo entre una fecha y hora inicial y final, generar tráfico equivalente al de diferentes

servicios, permite el envío de paquetes a partir de un fichero con los intervalos de transmisión de paquetes y el tamaño, permite emular comunicaciones cuando los tiempos y tamaños de paquete no son constantes y nos garantiza poder repetir las pruebas en las mismas condiciones.

2.3 Virtualización

La virtualización presenta una serie de ventajas, para algunos autores [9], [11] y [12], a la hora de realizar emulaciones de redes, dentro de ellas destacan la reducción de costes, espacio y recursos, además que permite una administración centralizada y simplificada por parte del investigador.

Por otro lado, en [21] se encuentra una descripción de las principales opciones de virtualización que algunos autores mencionan. En el ámbito de investigación es necesario que las máquinas virtuales consuman la mínima cantidad de recursos, que sean rápidas para poder adaptarse a cualquier escenario de prueba y que sean lo más cercano posible a una máquina real.

La interacción entre los sistemas operativos, tanto anfitrión como huésped, debe ser lo más eficiente posible para que el *software* que gestiona la interacción de cada huésped con el *hardware* no consuma recursos excesivos para la ejecución de ciertas instrucciones. En estos casos, es recomendable utilizar entornos de paravirtualización, en este ámbito, algunos autores [4], [7] y [22] han desarrollado sus investigaciones emulando escenarios de red en diversos contextos, para ello se han apoyado en Xen ya que en el grupo de investigación se utiliza este entorno de virtualización.

Xen presenta varias ventajas [23] frente a otro entornos de virtualización, entre ellos VMware ESXi, Hyper-V y KVM. Uno de los aspectos que destaca es que no hace utilización de *drivers* por lo que permite mantener aislados los sistemas huésped, además, al mantener los sistemas huésped aislado brinda cierto nivel de seguridad, así como los privilegios de acceso y la separación de los sistemas operativos.

2.4 Procesamiento matemático

La complejidad de algunas operaciones matemáticas o la gran cantidad de repeticiones de cálculo necesarias para el procesado de cierta información

hace necesario herramientas para este tipo de situaciones. Matlab es una herramienta de *software* que ofrece un lenguaje de programación (lenguaje M) para este tipo de ambientes. Es desarrollado por MathWorks desde el año 1984 y está disponible para las plataformas Unix, Windows y Apple Mac OS X.

A pesar de haber sido altamente difundido en universidades y centros de investigación y desarrollo en los últimos años, es un producto propietario de MathWorks y por lo tanto sujeto a licencia. Sin embargo, existen alternativas de libre distribución y código abierto como *GNU Octave* y *Scilab* que tienen buena compatibilidad con dicho lenguaje.

3 Metodología propuesta

3.1 Descripción de la metodología

La metodología utilizada en las pruebas realizadas se resume en cinco fases principales que se detallan a continuación:

- ***Fase 1:*** Estudio teórico y planificación de la prueba
- ***Fase 2:*** Acondicionamiento del entrono de pruebas
- ***Fase 3:*** Obtención de resultados
- ***Fase 4:*** Análisis de resultados
- ***Fase 5:*** Presentación de los resultados

Además, se puede revisar el apéndice B que contiene un procedimiento de tallado de las tareas realizadas para la obtención de modelos de tráfico IP para flujos multimedia.

3.1.1 Estudio teórico y planificación de la prueba

La planificación consiste en una etapa de adquisición de información a cerca del fenómeno que se desea analizar y de las herramienta y aplicaciones involucradas en el proceso. En concreto, en esta fase inicial se debe obtener toda la información del objeto de estudio que permita seleccionar cuales aplicaciones se deben elegir para obtener modelos de diversos flujos IP. Posteriormente, se debe realizar un listado de los elementos, dispositivo o equipos necesarios para la implementación de un escenario que involucre todos los aspectos de una red real y su correspondiente topología.

Dependiendo de los alcances y los objetivos de la investigación será necesario realizar una segregación de las tareas a realizar por grupos de trabajo, detallando los objetivos específicos y las tareas a realizar, además, es conveniente la elaboración de un cronograma de actividades que permita supervisar las actividades y tareas a realizar.

Por otro lado, las herramientas asociadas a la obtención de datos, captura de paquetes, así como, las relacionadas al análisis de la información deben ser

bien especificadas, ya que en algunos casos pueden que no existan o deben ser elaboradas por el investigador.

3.1.2 Acondicionamiento del entorno de pruebas

Unos de los aspectos más importantes a tener en cuenta es que la persona que realice las pruebas, debe tener un control total que permita la gestión de todos los nodos involucrados en el escenario a utilizar, contar con los privilegios administrativos correspondientes (administrador o *root*) según el sistemas operativos o de gestión utilizados, poder monitorizar los recursos del sistema y la posibilidad de realizar los cambios necesarios cuando se requieran.

La implementación de la topología planteado para realizar las pruebas sigue un orden bastante lógico, primero realizar la interconexión de los elementos de manera apropiada y en función de la topología planteada, seguidamente ponerlos en marcha. Con los equipos en funcionamiento, se realizan las configuraciones básicas de las interfaces de red que permitan la comunicación entre los diferentes nodos de la red y se comprueba el funcionamiento (con el comando *ping*, por ejemplo). Cuando los elementos de red se pueden comunicar, se realizan las configuraciones avanzadas (si es del caso) de las interfaces de los equipos terminales, o bien, aspectos relacionados a tipos de protocolo a utilizar, *codecs*, velocidades de transmisión de datos, niveles de compresión, entre otros.

En este punto, conviene realizar una monitorización de las interfaces utilizadas y de los procesos que se encuentran activos, ya que con esta información se pueden desactivar interfaces o eliminar procesos activos en el sistema que consuman recursos y que no sean necesarios, ya que algunas aplicaciones hacen un consumo de recursos mayor que otras (por ejemplo el tráfico de voz frente al de *streaming*), si esto no se hace, podría generar que el procesador no pueda entregar un paquete a la interfaz de red en el momento en que la aplicación lo solicite, generando un retardo en la transmisión que no corresponde a las características propias del tráfico que se esté modelando.

3.1.3 Obtención de resultados

La mayoría de las aplicaciones que brindan servicios IP deben ser configuradas en cierta medida, como por ejemplo, definir el tipo de protocolo que se utilizará para la transmisión en el caso de *streaming*, dirección destino o *broadcast*, número de puerto, entre otros. Todo este tipo de aspectos deben ser configurados antes de la captura de datos. Se puede hacer una prueba preliminar con la finalidad de comprobar el funcionamiento de todo el sistema en ejecución, si el sistema se ejecuta de manera normal se detiene la aplicación y se procede a la obtención de resultados.

Antes de lanzar de nuevo la aplicación para iniciar la captura de datos, se debe ejecutar y configurar el *sniffer* (algún tipo de filtro, por ejemplo), es conveniente que el *sniffer* sea un equipo externo a los nodos que intervienen en la comunicación con la finalidad de no cargar a dichos nodos con procesos ajenos al funcionamiento normal, por otro lado, es necesario que el equipo que realiza la captura de paquetes IP no inyecte tráfico a la red, que la única función que realice sea la de captura. Con todo esto preparado se inicia la transmisión de datos y se mantiene durante el tiempo que haya sido planificado, al término de dicho tiempo (o al final de la transmisión) se detiene la aplicación y luego se termina de realizar la captura de datos.

3.1.4 Análisis de resultados

Antes de realizar el análisis de los datos se debe corroborar que los paquetes capturados corresponden con el tráfico que se deseaba obtener, ya que si no es así, la prueba se debe repetir. Con la misma herramienta que se realizó la captura de paquetes se puede hacer una revisión rápida del tráfico obtenido, verificando que los encabezados de los paquetes correspondan con los protocolos utilizados por la aplicación que se quiere modelar, por ejemplo, que no se presente una gran cantidad de paquetes ICMP con avisos que el puerto es inalcanzable o para el caso de una aplicación de VoIP deben tener una cabecera de transporte UDP y no TCP, lo mismo para aplicaciones que utilicen RTP.

Aplicaciones como Wireshark y TCPDUMP permiten ciertos niveles de análisis o estadísticas, incluso generan gráficos estadísticos básicos en el caso de Wireshark, pero es usual que para ciertos tipos de análisis se requieran

herramientas más especializadas que el mismo investigador deba construir o adaptarlas a las ya existentes.

En este contexto, suele ser útil exportar los archivos de capturas realizadas a otro tipo de formato de archivo, por ejemplo archivos de texto, .csv, .xml, entre otros, que pueden editados o manipulados por *software* como hojas de cálculo, matlab, octave, scilab y otro tipo de aplicaciones que permiten un tratamiento matemático o de bases de datos más exhaustivo. También se debe tener en cuenta que muchos grupos de investigación a nivel mundial ponen a disposición ciertas herramienta que facilitan este tipo de labores, claro está, que esto depende del tipo de información que se esté manipulando, por lo que en la planificación se debe de realizar dicha búsqueda con la relevancia del caso.

Una vez realizados los análisis correspondientes o modelo de tráfico según sea el caso, estos deben ser comparados, en la medida de los posible, con resultados obtenido por otros investigadores que se hayan dedicado al estudio del mismo fenómeno, o en su defecto, que hayan generado pruebas muy parecidas, ya que la similitud entre dichos resultados dará evidencia de un procedimiento acertado y resultados válidos.

3.1.5 Presentación de los resultados

La forma en que los resultados de una investigación se presenta tiene un impacto importante hacia las personas a las que va dirigidas, además, que una buena manera de representar la información brinda a los lectores ideas claras y fáciles de entender. Para esto es necesario una selección de herramientas que permitan la reproducción de la documentación, fotografías, gráficos y demás elementos asociados. A continuación se comentan algunas de las principales herramientas utilizadas en el presente trabajo:

- **LaTeX:** Lenguaje que permite preparar automáticamente un documento de apariencia estándar y de alta calidad.
- ***LibreOffice:*** Conjunto de aplicaciones de productividad ofimática.
- ***PDFSam:*** Herramienta para dividir y fusionar documentos PDF.
- ***Bibus:*** Gestión de bases de datos de referencias bibliográficas.

- ***Inkscape:*** Editor de imágenes y gráficos vectoriales escalables.
- ***Gimp:*** Para edición de fotos e imágenes generales.
- ***Dia:*** Editor de diagramas, gráficos, diagramas de flujo, redes, etc.
- ***Planner:*** Gestor de proyectos, permite generar diagramas Gantt.
- ***Scilab:*** Herramienta matemática de alto nivel para el cálculo científico.
- ***Octave:*** Herramienta de análisis matemático con comandos similares a Matlab.

3.2 Adaptación de la metodología y automatización de procesos

Si el entorno de pruebas es WiFi se debe realizar un escaneo de los canales utilizados por las redes cercanas, con la finalidad de escoger un canal que perciba la mínima interferencia de las celdas adyacentes, por lo que cualquier otro tipo de entono de pruebas debe ser igualmente adaptado.

Para el caso de modelado de flujos IP multimedia, algunas herramientas deben ser elaboradas para poder realizar el análisis deseado, por ejemplo, en los anexos D.1 y D.2 se muestra un *script* que permite obtener ciertas estadísticas como paquetes enviados, recibidos, tiempos entre paquetes, tiempos entre ráfagas, medias y desviaciones estándar para los valores obtenido y gráficos.

Por otro lado, los datos procedentes de cierta aplicaciones no suelen ser compatibles con herramientas para el procesamientos de datos o la manipulación matemática, por esto, se deben de exportar o convertir a formatos que permitan el reconocimiento de la información, este es el caso de las capturas que se obtienen con TCPDUMP o Wireshark.

4 Modelado de flujos IP multimedia

4.1 Aplicaciones multimedia

Para realizar un análisis de diversos tipos de flujos de datos multimedia es necesario determinar y caracterizar los flujos IP de las aplicaciones que sean representativas del uso que los usuarios o empresas dan a Internet, o bien. En este ámbito, destacan tres clases de aplicaciones: voz, video y datos [4] y [8].

En este contexto, se encuentran aplicaciones de gran demanda por los usuarios como la radio y la televisión [4], el auge que ha tenido la telefonía por Internet, debido a la reducción de costes que esto conlleva y por otro lado la gran cantidad de cámaras de video, de las cuales hacen uso las empresas con fines de seguridad (televigilancia), las cuales se han trasladado al ámbito residencial, incluso. A continuación se describen algunas de dichas aplicaciones utilizadas en el presente trabajo.

En el caso de voz se utiliza como *gateway* VoIP un equipo Linksys SPA 3102 en cada uno de los extremos de la comunicación, cada uno con su respectivo teléfono convencional, para la interconexión de los dos nodos de red se utiliza un hub 3COM SUPER STACK II con una velocidad de 100 $Mbps$.

En las pruebas de vídeo para televigilancia realizadas en este trabajo se utiliza una cámara AXIS 2120 (un modelo diseñado para exteriores), en resumen, la cámara captura imágenes en formato JPEG y las transmite a razón de 25/30 tramas por segundo (PAL/NTSC) sobre una red con ancho de banda de 10 $Mbps$ o 100 $Mbps$ respectivamente.

Para proveer los servicios de vídeo *streaming* se utiliza el reproductor de multimedia VLC, la captura obtenida es de aproximadamente 180 s de la película *"Sintel"* [24], una película de animación 3D y de libre distribución. En la configuración de la aplicación de *streaming* se ha desactivado la transcodificación ya que consume mucho procesador, por lo se reproduce en el receptor en el mismo formato del archivo original (MP4), así el procesador no se satura y la calidad en la recepción es muy buena, además, se establece el uso de RTP y MPEG-TS (MPEG Transport Stream).

Para una información más detallada a cerca de las aplicaciones utilizadas, se puede consultar el apéndice C.

4.2 Obtención de modelos para flujos IP multimedia

Para realizar las estimaciones del los modelos de tráfico en flujos multimedia es necesario que el investigador tenga un control preciso del entorno de pruebas, minimizando los posibles errores e interferencia que puedan presentarse, dejando un escenario de pruebas que permita un desarrollo fluido a la aplicación que se dese modelar.

En las pruebas que se presentan a continuación, se ha seguido un orden específico, para cada una de ellas, en la manipulación de equipos y herramientas para la gestión de los datos, los pasos seguidos se pueden observar en el apéndice B. Para procesar las capturas se han realizado algunos *scripts* para la manipulación de los datos y la obtención de resultados estadísticos, los cuales pueden observarse en el anexo D.

4.2.1 Modelo de VoIP

En la figura 1 se muestra el diagrama utilizado para realizar las pruebas con la finalidad de obtener el modelo del tráfico en la transmisión de voz sobre IP.

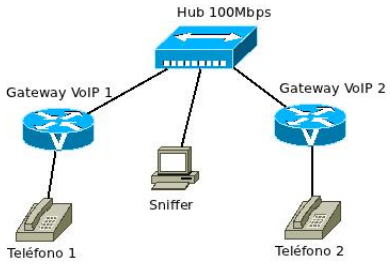


Figura 1: Escenario utilizado para determinar el tráfico de voz.

Después de la configuración de los equipos se lanza el *sniffer* para poder realizar la captura del tráfico, posteriormente, se realiza una llamada entre los terminales, mediante el *sniffer* se realizan capturas de paquetes mientras

la llamada está en curso y se analizan los resultados de dichas capturas.

VoIP utiliza el protocolo RTP para la transmisión de datos en tiempo real y este se transporta por medio de UDP, la figura 21 permite comprobar la utilización de los protocolos mencionados. Además, destaca en el análisis de las capturas de paquetes realizadas, que la transmisión de cada uno de los paquetes tiene una distribución constante (no se presentan ráfagas de paquetes), determinándose que cada terminal realiza el envío de paquetes cada $20\,ms$ y que el contenido de datos de dichos paquetes corresponde con el tipo de *codec* utilizado ($G.729$) y la cantidad de muestras que se definieron en la configuración de cada *gateway*.

Además, en la figura 2 se puede observar la variación de la duración de cada paquete para el tráfico enviado y recibido de voz en función del tiempo de transmisión. Para las muestras tomadas en este caso se presenta un tiempo medio de envío de paquetes de $20\,ms$ con una desviación estándar de $0.62\,ms$. Cada conexión de este flujo tienen un consumo de ancho de banda a nivel IP $BW = (60 \times 8)/20 \times 10^{-3}$, lo que equivale a 24 $Kbps$.

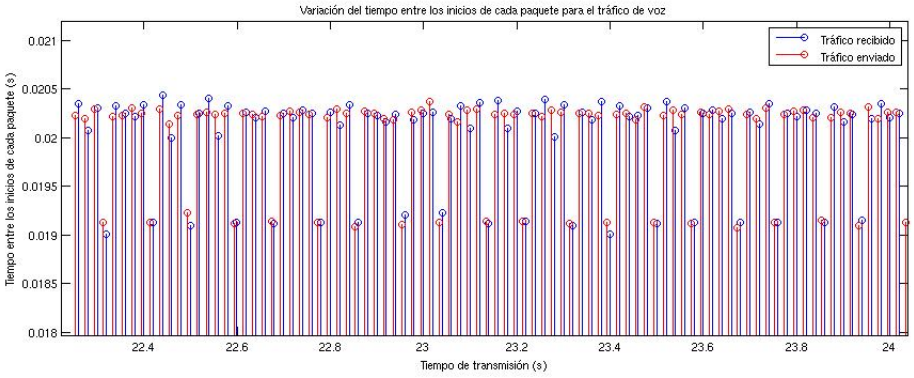


Figura 2: Detalle del tiempo entre los inicios de cada paquete para el tráfico de voz.

El modelo de la transmisión de voz es relativamente sencillo y estable en cuanto a tiempos y tamaños ya que no presenta un comportamiento de ráfagas, o bien, puede verse como una sola. Representar un flujo IP de este

tipo consiste en el envío de paquetes de 60 *byte* (incluyendo las cabeceras, ver figura 21) cada 20 *ms* con una desviación estándar de 0.62 *ms*, esto si se utiliza el *codec* *G*.729 y una paquetización de dos muestras por paquete. En el caso de que sea necesario el cambio del *codec* o la cantidad de muestras por paquete, los cambios son sencillos de representar, se debe tener en cuenta la frecuencia de muestreo definida en el nuevo *codec* utilizado y las muestras que se deseen empaquetar se incluirían después del encabezado RTP (ver figura 21).

4.2.2 Modelo de video para televigilancia

En este caso, la obtención del modelo de tráfico se realiza con un escenario como se muestra en la figura 3. La estación de trabajo puede tener acceso a la cámara IP (AXIS 2120) de manera remota por medio de un navegador web tradicional, la cámara utilizada puede ser configurada para un ancho de banda de 1 o 2 *Mbps*, según sea necesario dependiendo del enlace al que se tenga acceso, el *sniffer* capturará paquetes durante la transmisión de datos.

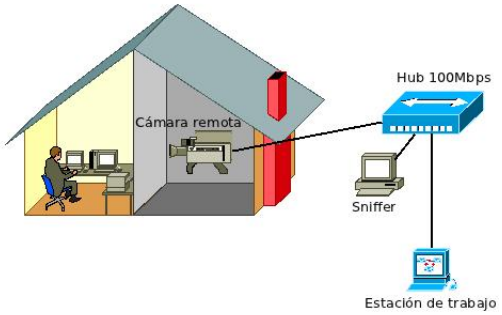


Figura 3: Escenario utilizado para determinar el tráfico de vídeo.

Para modelar este tipo de tráfico es necesario analizar aspectos más detallados de la configuración, ya que el comportamiento del flujo de datos difiere en función de la configuración que permita el fabricante para este tipo de aplicación, uno de los factores que tienen relevancia en este caso es el factor de compresión que se defina para la transmisión (ver tabla 2). Además de la influencia en la calidad de la imagen percibida por el usuario, el nivel de compresión que se defina afectará en forma directa el comportamiento del

flujo de paquetes en la red. Por último, otros aspectos se relacionan con el tamaño de la imagen (en *pixeles*) y el ancho de banda definido en la cámara.

El nivel de compresión define el tamaño de la imagen (en *bytes*) que la cámara transmite en cada instante, cada imagen tiene un tamaño diferente por lo que el último paquete de cada ráfaga, tendrá un tamaño menor que los demás, mientras que el resto serán de 1500 *bytes*, sin embargo, la cantidad de paquetes se mantiene constante para cada caso (excepto en uno de ellos), esto se aprecia en la tabla 2.

Resolución	*Nivel de compresión*	*Cantidad de paquetes*
704×576 *pixeles*	50 *Kbytes*	25
	16 *Kbytes*	10
352×288 *pixeles*	13 *Kbytes*	$7-9$
	4 *Kbytes*	3

Tabla 2: Cantidad de paquetes por ráfaga en función de la compresión para cámara.

El flujo de los paquetes se presenta en forma de ráfagas para todos los casos estudiados, cada ráfaga transmite la cantidad de paquetes que el nivel de compresión define (ver tabla 2), esto se muestra en la figura 4, donde el tiempo entre las llegadas de las ráfagas es el lapso en el cual se transmite la cantidad de paquetes; de forma análoga, existe un tiempo entre las llegadas de cada paquete contenido en la ráfaga que define la diferencia entre los inicios de cada paquete.

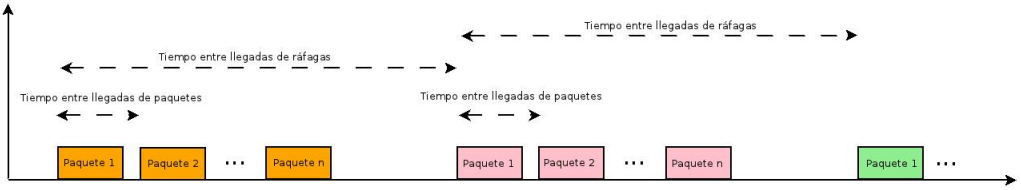


Figura 4: Relaciones de tiempos entre los inicios de cada paquete y de ráfagas para una compresión de 50 *Kbytes*.

Las capturas de datos se realizan para cada uno de los niveles de compresión que se muestran en la tabla 2, por otro lado, se estudia el caso de baja

compresión (50 $Kbytes$), ante cambios en el ancho de banda (permitido por el fabricante) definido en la configuración de la cámara. En este caso se analizan tres casos, éstos se muestran en la tabla 3, el primero, sin movimiento (Vídeo A), con poco (Vídeo B) y mucho movimiento (Vídeo C) frente a la cámara.

Tráfico	*Paquetes*		*Tiempo ráfagas (ms)*						
	Cantidad	*Tamaño (bytes)*	75	80	110	120	140	160	200
Vídeo A	10	1500	12.85&		47.35%		39.8%		
	1	500 – 800							
Vídeo B	24	1500		7.12%		30.65%		54.9%	9.29%
	1	700 – 1300							
Vídeo C	37	1500		42%	52%		11%	14.3%	4%
	1	200 – 1300							

Tabla 3: Resumen de los modelos obtenido para vídeo.

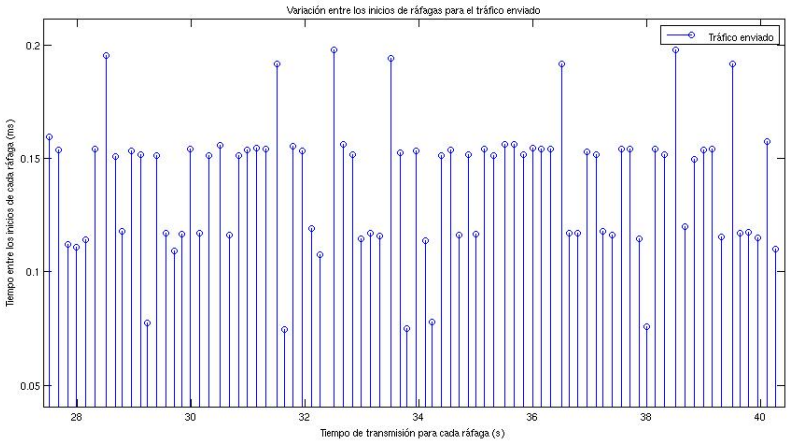


Figura 5: Detalle del tiempo entre los inicios de cada ráfaga para una compresión de 50 $Kbytes$.

En el caso donde no se presenta movimiento los diferentes tiempos entre los inicios de las ráfagas se mantienen en los valores que se muestran ($75\,ms$, $110\,ms$, $140\,ms$ aproximadamente), en los caso donde se presenta

movimiento los tiempos entre inicios de las ráfagas presentan mayores variaciones, esto se aprecia mejor en las figuras 6, 7 y 8.

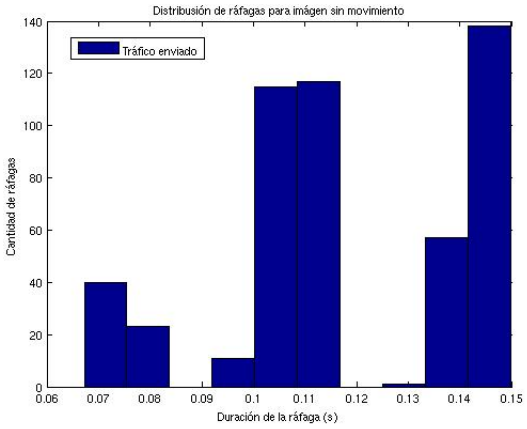


Figura 6: Distribución de ráfagas en función de su duración para una compresión de 50 $Kbytes$ sin movimiento.

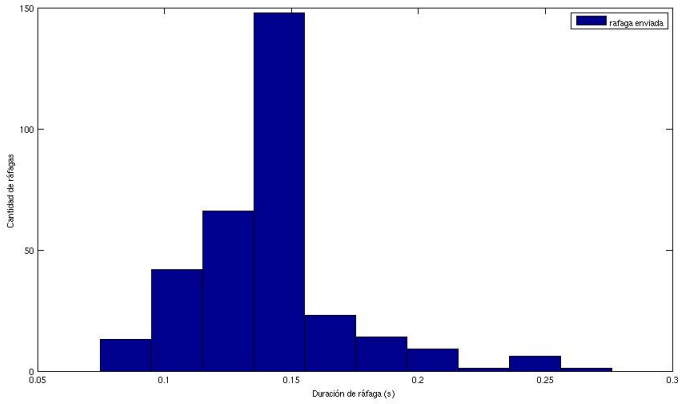


Figura 7: Distribución de ráfagas en función de su duración para una compresión de 50 $Kbytes$ con poco movimiento.

En la figura 5 se presentan los resultados correspondientes a la compresión 50 $kbytes$ y poco movimiento, en dicha figura se observa la duración entre

los inicios de cada paquete, además, muestra las variaciones de los tiempos entre los inicios de las ráfagas definidos en valores de 80 ms, 120 ms, 150 ms, 200 ms y 240 ms aproximadamente.

La distribución en función de la duración de cada ráfaga se muestra en las figuras 6, 7 y 8. Dichos histogramas permiten visualizar una clara agrupación de la duración entre los inicios de cada ráfaga. Los tiempos entre los inicios de cada paquete dependerán de la disponibilidad del procesador y la velocidad del enlace, sin embargo, se ha observado en el 1.32% de los casos para las transmisiones con movimiento moderado, que el último paquete de cada ráfaga presenta una duración de 40 ms.

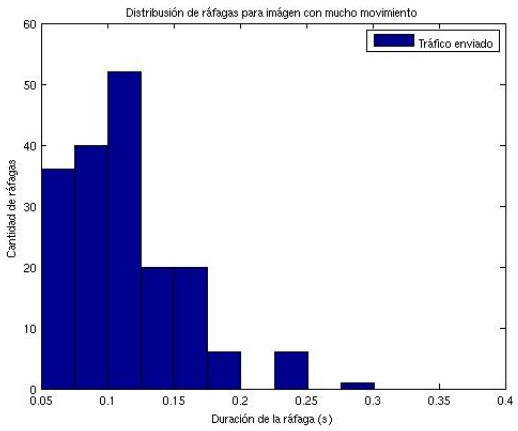


Figura 8: Distribución de ráfagas en función de su duración para una compresión de 50 $Kbytes$ con mucho movimiento.

El tráfico de vídeo se caracteriza por una agrupación de paquetes dentro de un determinado rango de tiempo definido o ráfaga. Para modelar el tráfico de vídeo que transmite la cámara se debe generar un archivo que tenga como contenido los tamaños y tiempos de transmisión de cada paquete, según la probabilidad de que un paquete esté contenido en un ráfaga determinada (ver figuras 6, 7 y 8 y tabla 3), ya que dentro de las características analizadas para este tipo de flujo, lo tamaños de todos los paquetes no son iguales, además, las variaciones en los tiempos entre ráfagas son bastante amplias, por lo que es necesario obtener una distribución estadística de dichos tiempos.

En el apéndice D.4 y en el D.5 se muestra como generar los archivos de texto correspondientes a los modelos de vídeo para una compresión de 4 $Kbytes$ y 50 $Kbytes$ respectivamente. En resumen, se puede decir que dichos *script* representan los histogramas de las ráfagas de cada una de las capturas correspondientes (ver figuras 6, 7 y 8), además de los tamaños y tiempos entre los inicios de cada paquete.

4.2.3 Modelo de *streaming*

Para modelar el tráfico *streaming* se utiliza un escenario como el que de presenta en la figura 9.

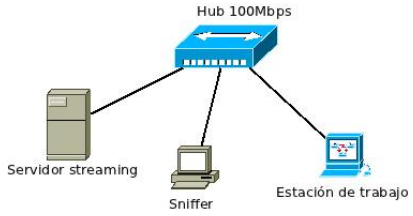


Figura 9: Escenario utilizado para determinar el tráfico streaming.

Analizando las capturas realizadas, se determina que el tráfico *streaming* no tiene un comportamiento uniforme, las ráfagas tienen grandes diferencias en cuanto a los tiempos entre los inicios entre ellas, como se puede apreciar en la figura 10. Las ráfagas producidas durante la transmisión presentan variaciones en la duración de las mismas, sin embargo, es claro que los paquetes tienen un tamaño de 1370 $bytes$, las ráfagas se caracterizan por mantener un periodo de inactividad antes del inicio de la próxima ráfaga, los tiempos entre los inicios de ráfagas difieren, sobrepasando en algunos hasta 1.5 s. El tamaño de los paquetes es el mismo en todos los casos, como era de esperarse por el uso de *Transport Stream*, ya que cada *stream* elemental tiene un tamaño fijo de 188 $bytes$ [25] y los paquetes IP deben contener múltiplos de éstos, por lo tanto el mayor tamaño posible sería de 1370 $bytes$.

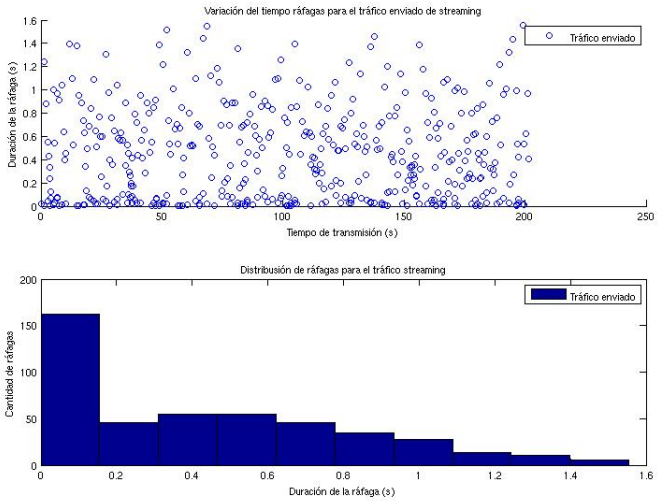


Figura 10: Distribución de tráfico transmitido por el servidor.

El modelo de *streaming* difiere en buena medida de los dos casos anteriores, éste se caracteriza por una gran dispersión en cuanto a los tiempos entre los inicios de las ráfagas. Sin embargo, la reproducción del tráfico se puede realizar de manera similar que con el vídeo, generando un archivo similar al mencionado anteriormente, pero cambiando en el *script* D.5 las ráfagas que se han definido y agregando nuevas ráfagas y duraciones de ráfagas en función de la distribución que se presenta en la figura 10, en este caso se recomienda comparar los histogramas que representan la distribución de las ráfagas generadas con las capturadas para corroborar que dichos histogramas sean similares, como se muestra al final de dicho *script*.

5 Modelado del *buffer* en un punto de acceso

5.1 Dimensionado de *buffer*

Internet es un conjunto descentralizado de redes de comunicación, de tal manera que la arquitectura es bastante heterogénea. Los nodos de red difieren en cuanto a capacidad, por mencionar un caso, las velocidades de entrada y salida de un *router* pueden tener grandes diferentes, mientras de manera interna, una red doméstica, puede tener una velocidad de hasta 54 *Mbps* (con un punto de acceso WiFi), la conexión hacia su proveedor de acceso a Internet puede ser de 8 *Mbps* (caso común en ADLS), el mismo caso se presenta en la interconexión de grandes ISP (*Internet Service Providers*) o en Puntos de Intercambio de Internet (*Internet eXchange Point*) con tasas de velocidad mayores. Otro de los posibles escenario es la conexión entre dos redes LAN de 100 *Mbps* por medio de un enlace inalámbrico de 54 *Mbps*, este caso se analizará con mayor detalle más adelante.

Estas diferencias entre las velocidad de entrada y de salida producen cuellos de botella donde se pueden producir pérdidas de paquetes. Los *router* utilizan *buffer* para reducir las pérdidas de paquetes absorbiéndolos cuando estos no pueden ser reenviados en ese preciso instante, también, se utilizan como instrumentos que ayudan a mantener los enlaces con un alto grado de utilización en casos de congestión.

Desde 1994, en [26] se propuso la denominada *rule of thumb*, la cual fue aceptada por muchos investigadores, para establecer el tamaño de los *buffer* en los *router*. La idea principal era la utilización al 100% del ancho de banda, asegurando una cantidad de paquetes almacenados que pueda mantener ocupado el canal cuando inicie el descarte de paquetes, mientras, TCP reacciona bajando la tasa de transmisión.

Esta regla, se resume en la ecuación 1, la cual define el tamaño del *buffer*, B, como el producto del ancho de banda del enlace, C, por el retardo de ida y vuelta, RTT.

$$B = C \times RTT \tag{1}$$

La ecuación 1 se obtuvo utilizando 8 flujos TCP en un enlace de 40 *Mbps*,

que en la actualidad no son datos representativos del tráfico en una red, por ejemplo, una capacidad de 40 $Gbps$, y un RTT de 250 ms, se obtendría un tamaño del *buffer* de 1.25 $Gbytes$ que es un tamaño muy grande, además, no se tomó en cuenta el caso de flujos con RTT diferentes.

En 2004, esta regla fué puesta en duda, por el llamado *Stanford model* [27], que reduce el tamaño del *buffer*, dividiéndolo por la raíz cuadrada del número N de flujos TCP, como se muestra en la ecuación 2. Esto se debe a que la ausencia de sincronización entre los flujos permite realizar una aproximación.

$$B = \frac{C \times RTT}{\sqrt{N}} \tag{2}$$

Esta nueva aproximación se realiza bajo el supuesto de que la duración de los flujos es larga y el número de flujos es lo suficientemente grande como para considerarlos asíncronos e independientes. Usando esta aproximación, un *router* que gestione 10.000 flujos solamente necesitaría 12.5 $Mbytes$ de tamaño de *buffer*. A este modelo se ha denominado *small buffer* [28].

Debido al modelo propuesto por [27] se generaron una serie de investigaciones en este ámbito. En [29] se propuso la utilización de *buffer* todavía más pequeños, denominados *tiny buffer*, que consideran que un tamaño de entre 20 y 50 paquetes (que equivale a algunas decenas de $Kbytes$) es suficiente como para alcanzar una utilización de entre el 80% y el 90%. Esto, basado en el hecho que los flujos no están sincronizados y el tráfico no presenta ráfagas, sin embargo, muchos de los flujos IP en tiempo real son definidos por un comportamiento de ráfagas como por ejemplo el *streaming* de vídeo, esto deja un poco de incertidumbre en cuanto a los modelos de dimensionado de *buffers*.

Por otro lado, son poco los trabajos realizados teniendo en cuenta servicios de tiempo real, que es uno de los principales objetivos del presente trabajo. En [1], [30] han considerado el tráfico combinado TCP y UDP en *buffer* pequeños, descubriendo una región anómala (ver figura 11), en la que las pérdidas de paquetes de UDP crecen con el aumento del tamaño del *buffer*.

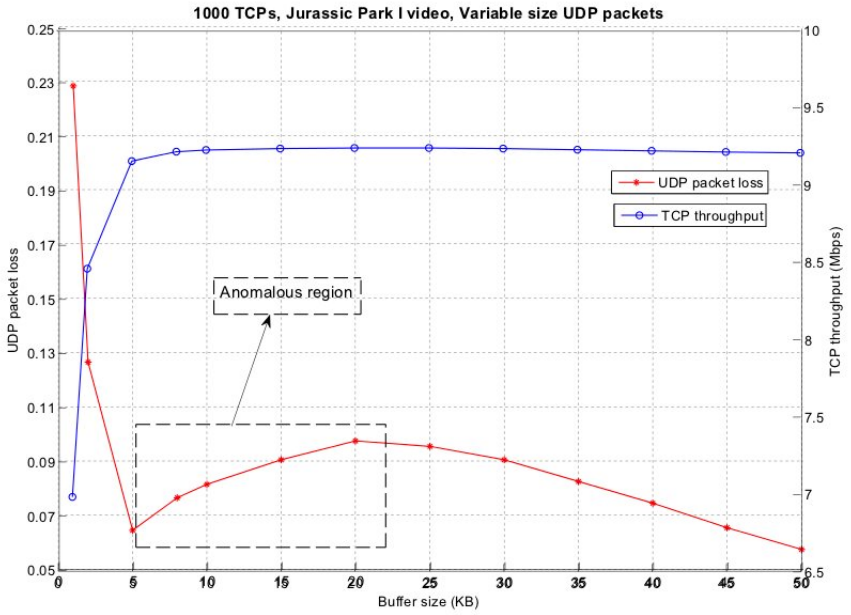


Figura 11: Pérdidas en paquetes UDP en función del tamaños de los *buffer* [1].

En [31] se presentó una simulación mediante NS2, con base en una topología en árbol con 18 nodos y enlaces de 50 *Mbps* de capacidad, que muestra las variaciones de la pérdida de paquetes en función del tamaño de los *buffer* para diferentes políticas de tráfico con la finalidad de mejorar el *Stanford model*.

Además, se ha notado que no existen unidades homogéneas entre los autores para referirse al tamaño de los *buffer*, ya que es común que se definan en términos de *bytes*, mientras que otros, lo hacen en paquetes, como se ha notado en [32].

Esto eventualmente podría generar cierta variación en el comportamiento, en casos donde el tamaño de los paquetes no es el mismo, que es un caso común cuando se mezclan diversos servicios que producen en la red paquetes de diferentes tamaños, desde unas decenas de *bytes*, hasta llegar al MTU máximo permitido por la red.

5.2 Modelado de *buffer*

La dimensión del *buffer* de un *router* no es un parámetro que forma parte de las especificaciones técnicas que un fabricante brinde de forma abierta, sin embargo, esta característica de diseño adquiere importancia en la planificación. Esto se debe a la relación que existe entre la utilización de un enlace y el tamaño del *buffer* en un *router*, por un lado, una excesiva cantidad de memoria generaría un incremento significativo en la latencia, ademas de un coste importante, en el caso opuesto, muy poca memoria para el *buffer* produciría un incremento en la pérdida de paquetes por el *router* en momentos de congestión.

Por esto, se ha planteado el escenario que se muestra en la figura 12, con él se determinarán las características más relevantes del *buffer* en el punto de acceso (AP1). Las estaciones de trabajo no poseen conexión inalámbrica, se conectan hacia los puntos de acceso por medio de un enlace de 100 *Mbps*, con lo que se garantiza que la comunicación entre las estaciones de trabajo estará limitado solamente por el enlace inalámbrico entre los puntos de acceso.

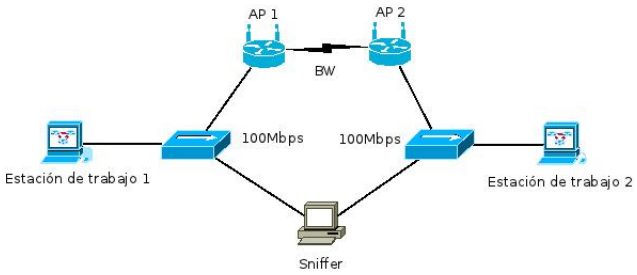


Figura 12: Escenario utilizado para determinar las características del *buffer* en un punto de acceso WiFi.

Las pruebas realizadas consisten en el envío de paquetes desde la estación de trabajo 1 hacia la 2, asegurando que la tasa de transferencia es lo suficientemente alta para congestionar el enlace inalámbrico. La captura de datos se realiza con un tercer equipo (*sniffer*) en dos puntos de la red, en transmisión y en recepción, de esta manera se asegura que las marcas de tiempo para cada paquete sean establecidas con la misma base de tiempo. Las pruebas se realizan para tres diferentes tamaños de paquetes ($300, 800$ y 1300 $bytes$)

y para diversos anchos de banda ($1, 2, 5.5, 11, 24$ *y* 54 $Mbps$).

La idea principal es escoger y mantener la tasa de transmisión con la finalidad de producir congestión en el enlace WiFi, con esto el *buffer* se saturará y por lo tanto, iniciará el descarte de paquetes por parte del *router*. Con este escenario se podría estimar el tamaño del *buffer*; teniendo en cuenta que un *router* sólo puede enviar un paquete a la vez, los paquetes que lleguen a dicho dispositivo mientras se envía uno de ellos, son almacenados en el *buffer*.

Como se ha mencionado con anterioridad, el *buffer* podría tener un tamaño definido en número de paquetes o *bytes*, para poder resolver esta interrogante se realiza cada prueba con flujos de diferentes tamaños de paquetes 300, 800 y 1300 *bytes*, de esta manera si el tamaño del *buffer* es el mismo en los tres casos, el tamaño estaría definido en número de paquetes, de lo contrario en *bytes*.

La estimación del tamaño se puede realizar de dos maneras diferentes, la primera sería estimarlo a partir del retardo de un paquete en la transmisión, la segunda, estimando el número de paquetes en cola en el momento que un paquete llega al receptor. La figura 13 muestra la manera que se ha seleccionado para determinar el tamaño del *buffer*. Cada paquete posee una marca de tiempo en el momento que es transmitido y otra cuando se recibe, en concreto, se contará la cantidad de paquetes que no han llegado al receptor en el periodo entre esos dos tiempos para cada uno de los paquetes.

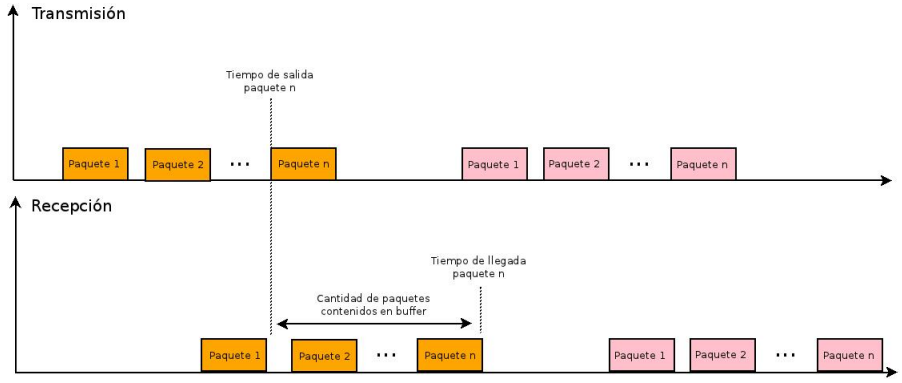


Figura 13: Diagrama de tiempos para la obtención del tamaño del *buffer*.

De forma experimental se ha determinado una tasa de transferencia de datos y se ha utilizado para todas las pruebas realizadas, dicha tasa consiste en el envío de paquetes cada 100 μs, estos valores se mantienen para todas las pruebas realizadas.

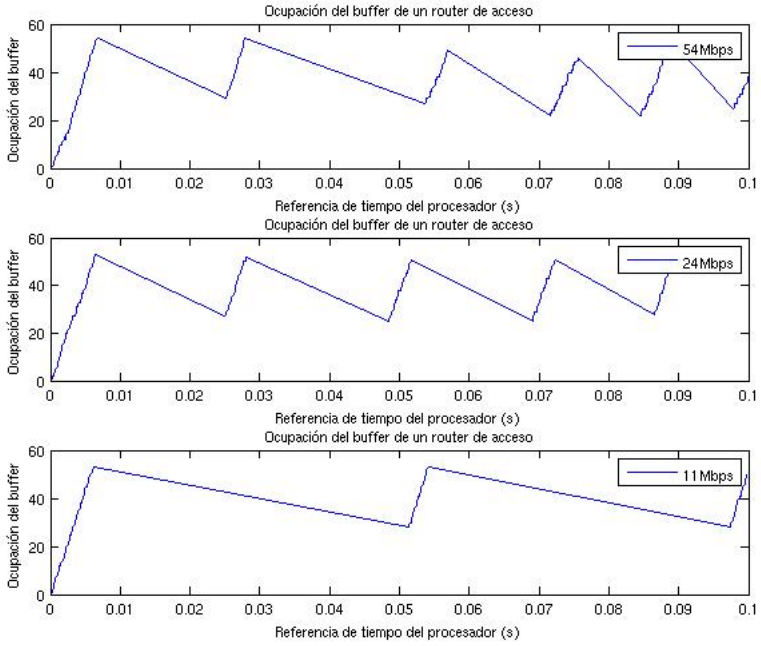


Figura 14: Ocupación del *buffer* de un *router* de acceso para un enlace de 54 *Mbps*, 24 *Mbps* y 11 *Mbps*.

Al realizar las pruebas y analizar los resultados, claramente se vé que la variación del tamaño de los paquetes no afecta a la cantidad de paquetes que el *buffer* puede almacenar, es decir, para los flujos de 300, 800 y 1300 *bytes* se ha estimado el mismo valor del tamaño del *buffer*, por lo tanto, se demuestra que el *buffer* analizado tiene un tamaño máximo definido en paquetes sin importar el tamaño de éstos y se estima en una cantidad de 55 paquetes como máximo. Al alcanzar el nivel de saturación del *buffer*, el *router* inicia el proceso de descarte de paquetes, sin embargo, cuando el *buffer* comienza a vaciarse no se admite el ingreso de nuevos paquetes hasta que éste alcance un valor de 30 paquetes en la cola como se aprecia en las figuras 14 y 15.

De esta manera se ha determinado que el *buffer* analizado posee dos niveles de umbral, uno de ellos define el tamaño máximo de éste, mientras que el otro se utiliza para permitir de nuevo el llenado del *buffer*. Por otro lado, las variaciones del ancho de banda para cada tamaño de paquetes no tiene efecto sobre dichos umbrales, es decir, el tamaño del *buffer* es el mismo en todo momento.

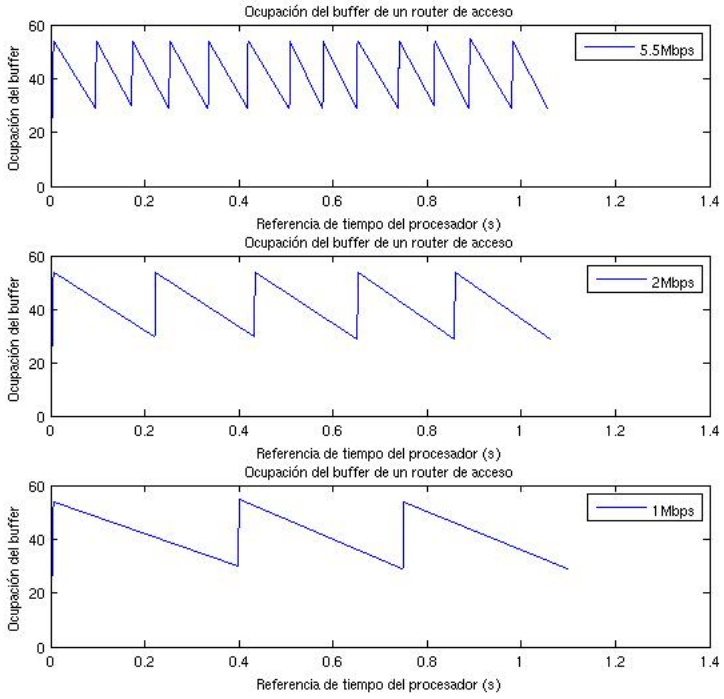


Figura 15: Ocupación del *buffer* de un *router* de acceso para un enlace de 5.5 *Mbps*, 2 *Mbps* y 1 *Mbps*.

En las figuras 14 y 15 se observa que la velocidad de llenado del *buffer* es mayor que la de vaciado en todos los casos, esto se debe a que los paquetes llegan al *router* por medio de un enlace *Ethernet* a 100 *Mbps*, mientras que el enlace de salida es donde se forma el cuello de botella ya que las tasas de transferencia en 802.11 b/g son menores. Al comparar las pendientes de las gráficas mencionadas, para los diferentes anchos de banda, se observa que

mientras mayor sea el ancho de banda, el tiempo de vaciado del *buffer* es menor, lo que supone que la latencia que experimentarán los paquetes será menor que en los casos donde el ancho de banda es mayor.

La velocidad de vaciado del *buffer* presentan ligeras variaciones a menores anchos de banda, sin embargo, cuando los puntos de acceso se configuran para anchos de banda más grandes el comportamiento difiere un poco de los demás casos, las velocidades de vaciado no se mantienen en valores similares como se puede observar en la tabla 4, además, se han observado casos donde el *buffer* no cumple con los valores de umbral mencionados anteriormente. Este comportamiento podría ser analizado más en detalle en futuras investigaciones.

	54 *Mbps*	24 *Mbps*	11 *Mbps*	5.5 *Mbps*	2 *Mbps*	1 *Mbps*
Tiempo (*ms*)	25.8	20.5	45.2	109.1	201.5	397.3
	8.8	14.2	43.4	83	191.7	380.8
Paquetes	27	27	25	24	24	25
	24	23	25	25	26	24
Velocidad (*Mbps*)	10.88	13.7	5.75	2.29	1.24	0.65
	28.36	16.84	6	3.13	1.41	0.65

Tabla 4: Variaciones observadas en las velocidades de vaciado del *buffer* en los diferentes anchos de banda.

6 Medidas de calidad

A continuación se presentan una serie de medidas basadas en entornos reales y con tecnología WiFi. Las medidas consisten en valorar la cantidad de conexiones que se pueden establecer sin pérdidas de datos, en los escenarios que se plantean más adelante. Los flujos IP utilizados para las pruebas son los modelos de voz y vídeo de televigilancia, que se han obtenido durante el presente trabajo.

Las pruebas presentadas se realizan tratando de mantener un máximo control de cada uno de los entornos de pruebas y de los equipos utilizados. Por otro lado, se ha realizado un escaneo de las celdas WiFi adyacentes que se encontraban operando a la hora de las pruebas, con la finalidad de seleccionar el canal que presente la menor interferencia para realizar las pruebas. El procedimiento que se ha utilizado se puede consultar en el apéndice B.

La generación de tráfico se realiza con la herramienta ETG, ya que ha sido desarrollada en el grupo de investigación donde se realiza el presente trabajo. Además, el tráfico generado es en todo momento UDP y en un solo sentido (*one way*), esto se detalla en la sección 6.5. La captura de paquetes es gestionada por el generador de tráfico mencionado, el cual hace uso de TCPDUMP con este fin. Además se verifica que cada una de las capturas sea correcta. Para el análisis y procesamiento de la información se utiliza Matlab.

Con la finalidad de tener un valor de referencia con el cual comparar los resultados obtenidos por las medidas, se ha realizado el cálculo teórico de la cantidad de conexiones que se pueden obtener en 802.11 en modo ad-hoc para diferentes anchos de banda. La tabla 5 muestra los cálculos obtenidos, suponiendo un enlace en modo ad-hoc, para el tráfico en un solo sentido (*one way*) y tráfico de ida y vuelta (*round trip*) en función del ancho de banda (ver apéndice C.4). Además, para cada caso se ha realizado el cálculo tomando como valores medios 0, 15.5 y 31 *slots* de *backoff*, lo que generaría una cantidad de conexiones mínima, media y máxima, respectivamente. La diferencia entre el tráfico de vídeo y voz se tienen en cuenta el ancho de banda y el tamaño de las tramas.

Tráfico	*BW*(*Mbps*)	*Cantidad de conexiones*					
		Round trip			*One way*		
		Mínimo	*Medio*	*Máximo*	*Mínimo*	*Medio*	*Máximo*
Voz	1	5	7	8	11	14	17
	2	7	10	14	15	20	29
	5.5	9	13	24	19	27	48
	11	10	15	30	20	31	60
	24	10	16	34	21	33	68
	54	11	17	37	22	34	74
Vídeo	1	0	0	0	1	1	1
	2	0	0	1	1	1	2
	5.5	2	2	2	4	4	5
	11	3	3	4	6	7	9
	24	4	6	8	9	12	17
	54	5	8	13	11	16	27

Tabla 5: Cantidad de conexiones de VoIP y video para diferentes anchos de banda. Para el caso de voz, se utiliza *codec* *G*729 [2] para una paquetización de dos muestras por paquete, para vídeo la longitud del paquete se estima en 1500 *bytes*.

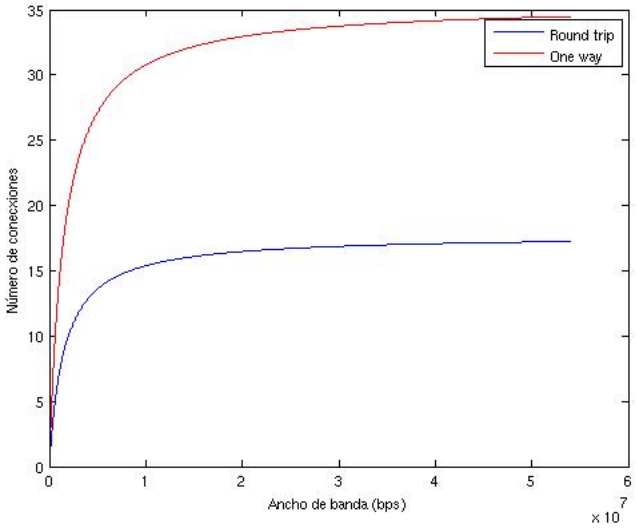


Figura 16: Cantidad de conexiones VoIP para un enlace WiFi en modo ad-hoc en función del ancho de banda del canal con un *backoff* de 15.5 *slots*.

La figura 16 muestra el crecimiento de la cantidad de flujos de voz en función

del ancho de banda de un canal para un *backoff* medio (15.5 *slots*). Es claro el rápido crecimiento, hasta un punto que el aumento de las conexiones se estabiliza. Por ejemplo, en el caso de un ancho de banda de 11 $Mbps$ se alcanzan 15 conexiones, pero si se aumenta el ancho de banda hasta 54 $Mbps$ (casi 5 veces más) sólo se ganan dos conexiones más para el caso de *round trip*.

6.1 Prueba en un enlace en modo ad-hoc

Para este caso se toman dos estaciones de trabajo con idénticas características de *hardware* y *software*. La configuración de la red (ver figura 17) se realiza por línea de comando en cada una de las estaciones, se comprueba el funcionamiento en modo *ad-hoc* y se realizan una serie de pruebas preliminares.

Figura 17: Enlace WiFi entre dos estaciones de trabajo en modo ad-hoc.

Con el modelo obtenido en la sección 4.2.1 se realizan pruebas de voz y progresivamente se aumenta la cantidad de conexiones en cada una de las pruebas hasta llegar al límite máximo permitido; que para el caso de voz es de 58 conexiones, por lo tanto, estos flujos tienen un consumo de ancho de banda de $BW = 58 \times (60 \times 8)/20 \times 10^{-3}$, lo que equivale a 1.392 $Mbps$. En la configuración de la tarjeta de red ($802.11b/g$), ésta gestiona de forma automática la norma (b/g) y por lo tanto el ancho de banda. El número de conexiones obtenido supera la media, pero no el máximo permitido, como se puede ver en la tabla 5.

Estos resultados suponen que el tiempo de *backoff* ha sido menor al valor medio esperado, o bien nulo, el protocolo de acceso al medio no hace uso del *backoff* o la cantidad de *slot* que fija es bastante pequeño.

Por otro lado, se realiza la misma prueba pero con tráfico de vídeo, el cual fue obtenido en la sección 4.2.2 y reproducido mediante el apéndice D.4,

para este caso se pudo obtener un máximo de 13 conexiones con la misma configuración y el ancho de banda sería $BW = 13 \times (3 \times 1500 \times 8)/40 \times 10^{-3}$, lo que equivale a 11.7 $Mbps$. El aumento de ancho de banda se debe a que se usan tramas de mayor tamaño.

6.2 Prueba en el enlace entre el punto de acceso y una estación de trabajo

En la figura 18 se muestra la segunda prueba realizada, en el enlace inalámbrico entre un punto de acceso y una de las estaciones base. Las medidas se basan en la premisa de que las posibles pérdidas se presentarán en el enlace inalámbrico ya que la conexión entre el punto de acceso y la otra estación de trabajo tiene una capacidad muy superior.

La configuración del punto de acceso se caracteriza por establecer una velocidad máxima de 1 $Mbps$ en modo 802.11g (para el primer caso, incrementándose en cada prueba), se inhabilita el modo CTS, el *beacon interval* es de 100 ms y se utiliza diversidad.

Figura 18: Enlace WiFi entre una estación de trabajo y un punto de acceso.

Además, se realizan pruebas con 802.11b obteniendo los mismos resultados para los casos donde los anchos de banda son los mismos que 802.11g (11 $Mbps$ o menos). La tabla 6 muestra los resultados obtenidos en relación con los valores teóricos calculados con un *backoff* medio (15.5 *slots*).

Los valores de voz obtenidos muestran que para los anchos de banda mayores 5.5 $Mbps$ (inclusive), la cantidad de conexiones supera las expectativas de los valores medios, pero siempre dentro de las posibles límites máximos.

Además, los resultados son comparables, a pesar de tener un escenario diferente, con otros estudios realizados por [5] y [12] en los que se obtienen los mismos resultados para el caso de 1 $Mbps$.

$Tráfico$	$BW(Mbps)$	$Cantidad\,de\,conexiones$				
		$Teórico$			$Experimental$	
		$Mínimo$	$Medio$	$Máximo$	$Conexinoes$	$BW(Mbps)$
Voz	1	11	14	17	12	0.288
	2	15	20	29	16	0.384
	5.5	19	27	48	32	0.768
	11	20	31	60	40	0.960
	24	21	33	68	40	0.960
	54	22	34	74	66	1.584
$Vídeo$	1	1	1	1	0	0
	2	1	1	2	1	0.9
	5.5	4	4	5	4	3.6
	11	6	7	9	8	7.2
	24	9	12	17	11	9.9
	54	11	16	27	13	11.7

Tabla 6: Cantidad de conexiones de VoIP y vídeo (*one way*) para diferentes anchos de banda. Para el caso de voz, se utiliza *codec* $G729$ para una paquetización de dos muestras por paquete, mientras que para vídeo se utiliza el tráfico obtenido en el apéndice D.4 y el escenario que se muestra en la figura 18. El ancho de banda experimental es a nivel IP.

Los resultados presentados en la table 6 son para el modelo de compresión de 4 $Kbytes$, dado que para el caso del modelo con una compresión baja (50 $Kbytes$) solamente permite una comunicación, habiendo pérdidas de paquetes desde la segunda comunicación.

Hay una clara diferencia en cuanto a cómo se aprovecha el ancho de banda para los casos de vídeo y voz; el protocolo de acceso al medio es el que hace que se penalicen más los paquetes pequeños que los de mayor tamaño, en cuanto a ancho de banda, ya que los tiempos de DIFS, SIFS y el tiempo del ACK se mantienen para todos los paquetes, así como el preámbulo.

6.3 Prueba en el enlace entre dos puntos de acceso

La siguiente prueba se realiza en un escenario como se muestra en la figura 19, en este caso cada estación de trabajo se conecta con un punto

de acceso diferente a través de un enlace de 100 *Mbps*, dichos puntos de acceso se comunican por medio de un enlace WiFi, las características de la configuración de los puntos de acceso se realiza de la misma manera que en la prueba anterior para diferentes anchos de banda.

Figura 19: Enlace WiFi entre dos puntos de acceso.

Tráfico	*BW*(*Mbps*)	*Cantidad de conexiones*				
		Teórico			*Experimental*	
		Mínimo	*Medio*	*Máximo*	*Conexiones*	*BW*(*Mbps*)
Voz	1	11	14	17	15	0.360
	2	15	20	29	28	0.672
	5.5	19	27	48	47	1.128
	11	20	31	60	56	1.344
	24	21	33	68	67	1.608
	54	22	34	74	68	1.632
Vídeo	1	1	1	1	0	0
	2	1	1	2	1	0.9
	5.5	4	4	5	4	3.6
	11	6	7	9	7	6.3
	24	9	12	17	11	9.9
	54	11	16	27	13	11.7

Tabla 7: Cantidad de conexiones de VoIP y vídeo (*one way*) para diferentes anchos de banda. Para el caso de voz, se utiliza *codec G*729 para una paquetización de dos muestras por paquete, mientras que para vídeo se utiliza el tráfico obtenido en el apéndice D.4 y el escenario que se muestra en la figura 19. El ancho de banda experimental es a nivel IP.

En este caso, se ha observado que la cantidad se flujos de voz que se han

podido establecer en cada ancho de banda, es mayor que los obtenidos en la prueba realizada anteriormente, como se puede observar en la tabla 8. Los resultados se mantienen dentro del rango medio de *slot* de que se han asumido, debido a las características del protocolo de acceso al medio. En el caso de vídeo los valores se mantienen similares a los obtenidos en la prueba anterior.

6.4 Prueba entre dos estaciones de trabajo en modo infraestructura

Esta prueba se realiza con dos estaciones de trabajo que se comunican por medio de un punto de acceso en modo infraestructura. La diferencia en este caso consiste en que ambas estaciones compartirán el medio por lo que la misma trama que se transmite aparecerá dos veces en el enlace WiFi, la configuración del punto de acceso es idéntica a los casos anteriores, el escenario propuesto se muestra en la figura 20.

Figura 20: Enlace WiFi entre dos puntos de acceso en modo infraestructura.

En este caso se presentan mayores problemas ya que a la hora de repetir las pruebas hay todavía más variaciones, es necesario una gran cantidad de repeticiones para poder obtener valores fiables en cuanto a sus valores medios. Por otro lado, recordar que se ha sido exhaustivo en mitigar los problemas de interferencias de las celdas adyacentes, sin embargo, los problemas asociados a las colisiones de datos generan problemas en la comunicación, degradando la cantidad de conexiones que se pueden obtener para cada ancho de banda en comparación a las pruebas anteriores.

Los datos que se muestran en la tabla 8 difieren en gran medida de los obtenidos en las pruebas anteriores, tanto para las pruebas de voz como para las de vídeo, encontrándose casos en los que la cantidad de conexiones se reduce hasta en un 57%.

Tráfico	*BW(Mbps)*	*Cantidad de conexiones*				
		Teórico			*Experimental*	
		Mínimo	*Medio*	*Máximo*	*Conexiones*	*BW(Mbps)*
Voz	1	11	14	17	11	0.264
	2	15	20	29	15	0.360
	5.5	19	27	48	22	0.528
	11	20	31	60	24	0.576
	24	21	33	68	50	1.200
	54	22	34	74	54	1.296
Vídeo	1	1	1	1	0	0
	2	1	1	2	1	0.9
	5.5	4	4	5	3	2.7
	11	6	7	9	5	4.5
	24	9	12	17	8	7.2
	54	11	16	27	7	6.3

Tabla 8: Cantidad de conexiones de VoIP y vídeo (*one way*) para diferentes anchos de banda. Para el caso de voz, se utiliza *codec* *G*729 para una paquetización de dos muestras por paquete, mientras que para vídeo se utiliza el tráfico obtenido en el apéndice D.4 y el escenario que se muestra en la figura 20. El ancho de banda experimental es a nivel IP.

6.5 Análisis de resultados

En primer lugar, los resultados obtenido para voz, se acercan más a los valores máximos teóricos que a los valores medios, esto se debe a que el entorno de pruebas se ha maximizado con esta finalidad, se ha seleccionado un canal que presente las mejores condiciones y se ha tratado de minimizar los efectos de las interferencias, además, se he tratado que la ubicación de los equipos utilizados asegure un excelente nivel de potencia, superior al 95% en todos los casos, con el objetivo de poder valorar el ancho de banda máximo en cada caso.

Por otro lado, a pesar que el número de *slots* se genera de manera aleatoria (dentro de ciertas condiciones), puede existir algún criterio dentro del

manejo que realiza la tarjeta de red, que al existir un buen nivel de señal y al no existir colisiones de datos con otras estaciones de trabajo, se reduzca el tiempo de *backoff*.

Como se pudo observar en 4.2.1, las comunicaciones de VoIP analizadas anteriormente, se caracterizan por paquetes pequeños, a esto se le suma la gran cantidad de encabezado que se genera en la red, que se puede calcular mediante la ecuación 3, de ahí se obtiene que el porcentaje de datos transmitidos con respecto a todos los bits transmitidos, lo que corresponde a un 21.27% en el caso de voz y un 95.21% para el de vídeo.

$$Porcentaje_{Datos} = \frac{Datos}{Datos + RTP + UDP + IP + MAC} \tag{3}$$

La relación anterior en conjunto con el *backoff*, son las grandes limitantes de CSMA/CA en cuanto a la cantidad de conexiones que puedes ser transportados por un canal determinado utilizando esta técnica de acceso al medio; por otro lado, el preámbulo que se utilice (largo o corto) también tiene un aporte significativo, además, se debe tener en cuenta los tiempos DIFS, SIFS y de ACK.

Los resultados obtenidos concuerdan con los valores teóricos calculados y con ciertos autores [5] y [12], lo que valida el uso de la metodología empleada para realizar las pruebas en los distintos escenarios planteados. Además, las herramientas utilizadas para la reproducción del tráfico han tenido los resultados esperados por lo que se recomienda su uso.

Para el caso de vídeo y voz el comportamiento de los flujos es similar en los distintos escenarios, excepto en modo infraestructura ya que al compartir las dos estaciones de trabajo el mismo medio, esto produce colisiones de datos, que puede provocar un aumento en el *backoff*.

Otro aspecto que es necesario dejar claro está relacionado con el tipo de tráfico y el sentido de los flujos IP analizados en esta sección, esto se debe en buena medida a la limitación de tiempo para el desarrollo del presente trabajo. Dejando planteado un análisis que contemple ambos sentidos de la comunicación para futuras investigaciones.

Otro de los aspecto tenidos en cuenta a la hora de establecer las características del tráfico utilizado en la realización de las medidas, viene dado por el análisis de los modelos de tráfico obtenidos, por ejemplo, en el caso del modelo de voz se pudo determinar que cada uno de los nodos enviaba tráfico idéntico al nodo con el cual se comunicaba, de aquí que una de las aproximaciones que se infieren sería que la cantidad de comunicaciones posibles en ambos sentidos sería la mitad de las obtenidos en un solo sentido, además, lo que se desea valorar en este caso es la cantidad de flujos posibles sin pérdidas. Sin embargo, si se deseara valorar el efecto real del tráfico en ambos sentidos habría que transmitir desde las dos estaciones de trabajo, tráfico con una distribución similar a la utilizada.

En el caso de vídeo, el análisis del modelo obtenido (para una compresión de $50\,Kbytes$) presenta un 25% de los paquetes en sentido inverso, dichos paquetes son del tipo ACK, ya que la transmisión se realiza por medio de TCP. Esta situación se puede ser aproximada mediante tráfico UDP si se supone que no hay pérdidas, que es el caso de estudio en la realización de las pruebas planteadas. Por otro lado, en futuras investigaciones, sería interesante valorar el efecto del aumento de flujos en ambos sentidos del enlace, para esto es necesario apoyarse en el *script* D.1 donde se obtienen las características del tráfico en ambos sentidos de la comunicación, de la misma manera comentada con anterioridad sería necesario reproducir el tráfico desde los dos equipos terminales según el sentido del flujo de datos.

6.6 Recomendaciones para medidas en entornos virtuales

La virtualización de escenarios de red presenta una serie de ventajas en comparación a la simulación; el uso de entornos virtuales permite utilizar un determinado sistema operativo y aplicaciones en forma directa, algo que es útil en los casos que éstos tengan efecto sobre el tráfico analizado con respecto al su de escenarios reales, por otros lado, presenta una reducción importante de costes cuando el número de nodos se incrementa.

En el caso de hacer uso de este tipo de entorno para la realización de pruebas, emulando entornos reales, se debe prestar especial cuidado a los dos primeros

puntos de la metodología planteada en este trabajo (ver apéndice B). Para construir un entorno de red virtual se debe seguir un orden lógico de pasos para poder representar de la mejor manera una red o parte de ésta, por ejemplo, se podría seguir un procedimiento como este:

1. Crear la máquina virtual o copiar una imagen a partir de una existente. Usualmente en una ruta como la siguiente: */extra/ape/xen-gtc*.

2. Adaptar el archivo de configuración de cada máquina virtual, si es del caso, agregar interfaces de red al dominio 0 y las demás que sean necesarias. Usualmente en una ruta como la siguiente: */etc/xen*.

3. Crear los *hub* virtuales que sean necesarios para la interconexión de la red.

4. Arrancar las máquinas virtuales.

5. Configurar cada interfaz de red (dirección, máscara, etc.) si es necesario.

6. Configurar aspectos avanzados como por ejemplo limitaciones de ancho de banda, tipos de *buffer*, entre otros.

Limitar el ancho de banda en una interfaz determinada es una manera de emular un enlace físico que puede transportar datos a cierta velocidad, ya que los enlaces a través de los *hub* virtuales estarían operando a una velocidad determinada por el nivel de ocupación del procesador, en linux esto se puede realizar mediante el uso de TC y en el caso de ser necesario la introducción de pérdidas y/o retardos se puede hacer uso de NETEM.

Los *buffer* también pueden ser introducidos en este ámbito, TC permite crear disciplinas de colas (*QDISCS*) para la gestión del tráfico. Las disciplinas de colas que se encuentran definidas en TC son:

$$QDISCS \begin{cases} Classless \begin{cases} [p/b]fifo \\ pfifo\_fast \\ red \\ sfq \\ tbf \end{cases} \\ Classfull \begin{cases} CBQ \\ HTB \\ PRIO \end{cases} \end{cases}$$

Algunos de los *buffer* sin clases (*classless*) como el *[p/b]fifo* son una cola *fifo* (*first in first out*) donde el tamaño se puede definir en paquetes o *bytes*, mientras otros como el *red* (*Random Early Detection*), definen dos umbrales, el máximo se utiliza para determinar el punto de inicio del descarte de paquetes, mientras que el mínimo ayuda a definir una banda (entre el mínimo y el máximo) dentro de la cual los paquetes pueden ser descartados de manera aleatorio con la probabilidad que se establezca.

Por otro lado, los *buffer* con clases (*classfull*), se basan en jerarquías y prioridades, como es el caso de *CBQ* donde dicha jerarquía se basa en el ancho de banda, otro caso es el de *PRIO*, donde los flujos se clasifican en prioridades y se crean colas *fifo* para cada prioridad y la cola con mayor prioridad es la primera en ser seleccionada para el envío de paquetes, hasta que dicha cola esté vacía, se inicia el envío de paquetes de la siguiente cola según el orden de prioridad de cada cola.

En la sección 5 se han encontrado las características necesarias para modelar el *buffer* de un punto de acceso en concreto con la finalidad de realizar medidas, sin embargo, el comportamiento de dicho *buffer* no forma parte de las disciplinas de colas que el kernel de linux permite gestionar por el momento. Además, por motivos del tiempo, no se ha podido realizar dicha implementación, quedando abierta una nueva línea de investigación en este ámbito.

7 Conclusiones y líneas futuras de investigación

7.1 Conclusiones

La metodología propuesta en la sección 3 ha sido utilizada para la realización de todas las medidas planteadas en el presente trabajo, obteniendo resultados que permitieron modelar diferentes tipos de flujos IP multimedia y características funcionales de *buffer*. El procedimiento utilizado se resume en el apéndice B.

La metodología utilizada para el desarrollo de las pruebas presentadas en este trabajo provee una técnica para el análisis y modelado de flujos IP. Los modelos así obtenidos, son representativos del comportamiento real de las diversas aplicaciones analizadas.

Los modelos encontrados difieren entre sí en cuanto al comportamiento del tráfico según el tipo de flujo. Para el caso de VoIP (con *codec* $G.729$) presenta características muy estables en cuanto al envío y el tamaño de los paquetes. Estas dos características son determinadas por la cantidad de muestras que contenga cada paquete.

Sin embargo, el comportamiento en televigilancia y *streaming* presentan distribuciones no uniformes. El modelo propuesto consiste en un envío de ráfagas de paquetes con tiempo entre ráfagas y tamaño de las mismas dado por una distribución estadística.

En cuanto al *buffer* analizado, éste posee dos niveles de umbral. Uno de ellos define el tamaño máximo (55 paquetes) a partir de este valor inicia el descarte de paquetes, mientras que el otro, el mínimo (30 paquetes) permite nuevamente el llenado del *buffer*. Las pruebas se han realizado de tal forma que la velocidad de llenado del *buffer* sea mayor que la de vaciado y así se consigue una situación de congestión que permita su estudio. Las velocidades de vaciado del *buffer* varían para los diferentes anchos de banda de WiFi. Se observa que al configurar el punto de acceso a velocidades altas, dicha velocidad de vaciado presenta grandes variaciones (ver tabla 4). En cuanto el ancho de banda de WiFi va disminuyendo se presentan algunas

variaciones pero no tan relevantes como en los otros casos.

Las medidas realizadas en la sección 6 para los cuatro escenarios planteados, muestran la dependencia que tiene el proceso de *backoff* del ancho de banda útil de un enlace inalámbrico. El proceso de *backoff*, a su vez, depende de los posible problemas de interferencias, colisiones de datos en el medio e incluso del fabricante. El ancho de banda obtenido a nivel IP para las pruebas realizadas en dicha sección, muestra que dadas las características del protocolo de acceso al medio, las aplicaciones que generan paquetes con tamaños pequeños transmitirán menor cantidad de *bytes* en un tiempo determinado, en comparación con aplicaciones que generen paquetes de tamaños mayores. Esto se debe a que los tiempos de DISF, SIFS y del ACK se mantienen tanto para los paquetes grandes como para los de menor tamaño, en cuyo caso se presentan pérdidas de eficiencia.

Por último, las pruebas realizadas en modo infraestructura presentan una menor cantidad de flujos posibles en los diferentes anchos de banda estudiados, debido a que una trama estará presente en ambos sentidos del enlace. Esto también sucederá cuando dos o más estaciones acceden al mismo medio.

7.2 Líneas futuras de investigación

Los escenarios planteados a lo largo del trabajo son casos de uso común. Sin embargo, en futuras investigaciones se puede ampliar el número de casos de uso con nuevos escenarios de red, estudio de diferentes *buffer* tanto en puntos de acceso como en *router* de acceso y otro tipo de aplicaciones, por ejemplo videoconferencia. Además, se podría valorar los efectos en ambientes con mayor nivel de interferencia.

Las medidas realizadas en el presente trabajo se basan en el análisis de un sentido de la comunicación. Se ha observado que algunos de los flujos multimedia presentan paquetes en sentido inverso, por lo que se deja planteado para una futura investigación el análisis del tráfico en ambos sentidos del enlace. Dicho estudio puede tomar como referencia los modelos observados en el presente trabajo.

Por otro lado, la generación de datos para el caso de vídeo se ha realizado

mediante UDP, con base en la premisa de que no habría retransmisiones, ya que las mediciones realizadas se basan en el número de conexiones posibles sin pérdidas, y además, que no se ha generado el flujo de ACK en sentido inverso. Por este motivo, sería importante complementar este trabajo con una investigación que valore el impacto del flujo de ACK en sentido inverso para este tipo de tráfico.

En el presente trabajo se pudo determinar algunas de la características técnicas y funcionales del *buffer* en un punto de acceso específico. Sin embargo, el tiempo ha sido una limitación para poder investigar aspectos más relacionados con el comportamiento de los *buffer* ante diferentes tipos de flujos IP en tiempo real y otros tipos de equipos y fabricantes.

Bibliografía

[1] Arun Vishwanath and Vijay Sivaraman. Routers With Very Small Buffers: Anomalous Loss Performance for Mixed Real-Time and TCP Traffic. pages 80–89, June 2008.

[2] Telecomunication Standardization Sector of ITU. H.222.0 infrastructure of audiovisual services – transmission multiplexing and synchronization. Technical report, International Telecomunication Union, 2006.

[3] Padmavathi Mundur and Poorva Arankalle. Optimal server allocations for streaming multimedia applications on the internet. *Computer Networks*, 50(18):3608 – 3621, 2006.

[4] J. Ruiz Mas, J. I. Aznar Baranda, J. M. Saldaña Medina, J. Fernández Navajas, B. Hernández Ortega, L. Blasco Arcas, and J. Jiménez Martínez. Evaluación de nuevos canales de distribución en servicios interactivos ip. *IX Jornadas de Ingeniería Telemática (JITEL 2010) XX Jornadas Telecom I+D*, Septiembre 2010.

[5] 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 managed ip telephony system. *Proc. New Technologies, Mobility and Security (NTMS), 5th International Conference*, Paris. ISBN: 978-1-4244-8704-2. Febrero 2011.

[6] Assen Golaup and Hamid Aghvami. A multimedia traffic modeling framework for simulation-based performance evaluation studies. *Computer Networks*, 50(12):2071 – 2087, 2006. Network Modelling and Simulation.

[7] J. M. Saldaña Medina, Jenifer Murillo Royo, J. Fernández Navajas, J. Ruiz Mas, J. I. Aznar Baranda, and Eduardo Viruete Navarro. Bandwidth efficiency improvement for online games by the use of tunneling, compressing and multiplexing techniques. *Proc. International Symposium on Performance Evaluation of Computer and Telecommunication Systems SPECTS*, pages 227–234, The Hague, Netherlands. ISBN: 978-161-782-309-1. Junio 2011.

[8] Nelson Antunes, António Pacheco, and Rui Rocha. An integrated traffic model for multimedia wireless networks. *Computer Networks*, 38(1):25 – 41, 2002.

[9] 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.

[10] J. I. Aznar Baranda, E. A. Viruete Navarro, J. Fernández Navajas, J. Ruiz Mas, J. M. Saldaña Medina, and J. Murillo. Qmoes: A bandwidth estimation and monitoring tool for qoe-driven broadband networks. *In Proc. New Technologies, Mobility and Security (NTMS), 5th International Conference*, Paris. ISBN: 978-1-4244-8704-2. Febrero 2011.

[11] J. M. Saldaña Medina, J. I. Aznar Baranda, E. A. Viruete Navarro, J. Fernández Navajas, and J. Ruiz Mas. Qos measurement-based cac for an ip telephony system. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 22(1):3–19, DOI: 10-1007-978-3-642-10625-5-1. Noviembre 2009.

[12] J. Murillo Royo, J. M. Saldaña Medina, 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.

[13] J. M. Saldaña Medina, J. Murillo, J. Fernández Navajas, J. Ruiz Mas, E. A. Viruete Navarro, and J. I. Aznar Baranda. Evaluation of multiplexing and buffer policies influence on voip conversation quality. *In Proc. CCNC 2011 3rd IEEE International Workshop on Digital Entertainment, Networked Virtual Environments, and Creative Technology*, pages 1147–1151, Las Vegas. ISBN 978142448782. Enero 2011.

[14] J. M. Saldaña Medina, 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.

[15] http://www.tcpdump.org/.

[16] http://www.wireshark.org/.

[17] http://www.linuxfoundation.org/collaborate/workgroups/networking/netem.

[18] http://www.cs.helsinki.fi/u/jmanner/software/.

[19] http://www.grid.unina.it/software/ITG/.

[20] Casadesus Pazos L. A., Fernández Navajas J., Ruiz Mas J., Saldaña Medina J. M., Aznar Baranda J. I., and Viruete Navarro Eduardo. Herramienta para automatización de medidas de tiempo real extremo a extremo. *Actas del XXVI Simposium Nacional de la Unión Científica Internacional de Radio (URSI 2011)*, Leganés (España). ISBN 9788493393458. Septiembre 2011.

[21] Saldaña J. Sistema de emulación de escenarios de movilidad. Master's thesis, Universidad de Zaragoza, Agosto 2008.

[22] Murillo J. Análisis de un sistema cac par telefonía ip. Master's thesis, Universidad de Zaragoza, Febrero 2010.

[23] http://www.xen.org/.

[24] http://www.sintel.org/.

[25] Telecomunication Standardization Sector of ITU. G.729 coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (cs-acelp). Technical report, International Telecomunication Union, 2007.

[26] Curtis Villamizar and Cheng Song. High performance tcp in ansnet. *SIGCOMM Comput. Commun. Rev.*, 24:45–60, October 1994.

[27] Guido Appenzeller, Isaac Keslassy, and Nick McKeown. Sizing router buffers. *SIGCOMM Comput. Commun. Rev.*, 34:281–292, August 2004.

[28] Arun Vishwanath, Vijay Sivaraman, and Marina Thottan. Perspectives on router buffer sizing: recent results and open problems. *SIGCOMM Comput. Commun. Rev.*, 39:34–39, March 2009.

[29] Mihaela Enachescu, Yashar Ganjali, Ashish Goel, Nick McKeown, and Tim Roughgarden. Part iii: routers with very small buffers. *SIGCOMM Comput. Commun. Rev.*, 35:83–90, July 2005.

[30] A Vishwanath, V Sivaraman, and G N Rouskas. Considerations for sizing buffers in optical packet switched networks. *IEEE INFOCOM 2009 The 28th Conference on Computer Communications*, pages 1323–1331, 2009.

[31] Amogh Dhamdhere and Constantine Dovrolis. Open issues in router buffer sizing. *SIGCOMM Comput. Commun. Rev.*, 36:87–92, January 2006.

[32] Joel Sommers, Paul Barford, Albert Greenberg, and Walter Willinger. An sla perspective on the router buffer sizing problem. *SIGMETRICS Perform. Eval. Rev.*, 35:40–51, March 2008.

[33] http://www.videolan.org/.

[34] IEEE Standard for Information Tecchnology. Wireless lan medium access control (mac) and physical layer (phy) specifications. Technical report, IEEE Computer Society, 2007.

A Acrónimos y términos

A.1 Acrómimos

GNU GPL (*GNU General Public License*)
NETEM *Network Emulator* es un emulador de red.
TC (*Traffic Control*)
JTG (*Jugi's Traffic Generator*)
GUI (*Graphical User Interface*)
D-ITG
OWD (*one-way-delay*)
RTT (*round-trip-time*)
IAX (*Inter-Asterisk eXchange protocol*)
Hub Concentrador o ethernet hub, un dispositivo para compartir una red de datos o de puertos USB de un ordenador.
IVR (*Interative Voice Response*)
NAT (Network Address Translation - Traducción de Dirección de Red) es un mecanismo utilizado por enrutadores IP para intercambiar paquetes entre dos redes
CCITT Consultative Committee for International Telegraph and Telephone (Comité Consultivo Internacional de Telefonía y Telegrafía)
H.323 Estándar de la ITU-T para voz y videoconferencia interactiva en tiempo real en redes de área local, LAN, e Internet.
IP Internet Protocol (Protocolo Internet)
ISP Internet Service Provider (Proveedor de Servicios Internet, PSI)
ITU-T International Telecommunications Union Telecommunications (Unión Internacional de Telecomunicaciones - Telecomunicaciones)
MOS Mean Opinion Score (Nota Media de Resultado de Opinión)
QoS Quality of Service (Calidad de Servicio)
RTP Real Time Protocol (Protocolo de Tiempo Real)
SIP Session Initiation Protocol (Protocolo de Inicio de Sesión)
TCP Transmission Control Protocol (Protocolo de Control de Transmisión)
UDP User Datagram Protocol (Protocolo de Datagramas de Usuario)
ICMP El Protocolo de Mensajes de Control de Internet o ICMP (por sus siglas de Internet Control Message Protocol) es el sub protocolo de control y notificación de errores del Protocolo de Internet (IP).
MPEG Moving Picture Experts Group (en español Grupo de Expertos en Imágenes Móviles), referido comúnmente como MPEG, es un grupo de trabajo del ISO/IEC encargado de desarrollar estándares de codificación de audio y vídeo.

A.2 Términos

codec Algoritmo software usado para comprimir/descomprimir señales de voz o audio. Se caracterizan por varios parámetros como la cantidad de bits, el tamaño de la trama (frame), los retardos de proceso, etc. Algunos ejemplos de codecs típicos son G.711, G.723.1, G.729 o G.726.
streaming consiste en la distribución de audio o video por Internet, esta palabra hace referencia a una transmisión en forma continua

gateway (pasarela). Dispositivo empleado para conectar redes que usan diferentes protocolos de comunicación de forma que la información puede pasar de una a otra. En VoIP existen dos tipos principales de pasarelas: la Pasarela de Medios (Media Gateways), para la conversión de datos (voz), y la Pasarela de Señalización (Signalling Gateway), para convertir información de señalización.

jitter (variación de retardo). Es un término que se refiere al nivel de variación de retado que introduce una red. Una red con variación 0 tarda exactamente lo mismo en transferir cada paquete de información, mientras que una red con variación de retardo alta tarda mucho más tiempo en entregar algunos paquetes que en entregar otros. La variación de retardo es importante cuando se envía audio o video, que deben llegar a intervalos regulares si se quieren evitar desajustes o sonidos inintelegibles.

sniffer un analizador de paquetes es un programa de captura de las tramas de una red de computadoras.

router (encaminador, enrutador). Dispositivo que distribuye tráfico entre redes. La decisión sobre a donde enviar los datos se realiza en base a información de nivel de red y tablas de direccionamiento. Es el nodo básico de una red IP.

VoIP Voice over IP (Voz sobre IP). Método de envío de voz por redes de conmutación de paquetes utilizando TCP/IP, tales como Internet.

Ancho de banda Capacidad de transmisión de datos que tiene un medio determinado, generalmente cuantificado según el número de bits que se transmiten en un segundo.

root En sistemas operativos del tipo Unix, root es el nombre convencional de la cuenta de usuario que posee todos los derechos en todos los modos (mono o multi usuario). root es también llamado superusuario. Normalmente esta es la cuenta de administrador.

broadcast transmisión de un paquete que será recibido por todos los dispositivos en una red. El Dominio de difusión,

CSMA/CA Carrier Sense, Multiple Access, Collision Avoidance (acceso múltiple por detección de portadora con evasión de colisiones) es un protocolo de control de redes de bajo nivel que permite que múltiples estaciones utilicen un mismo medio de transmisión. Cada equipo anuncia opcionalmente su intención de transmitir antes de hacerlo para evitar colisiones entre los paquetes de datos (comúnmente en redes inalámbricas, ya que estas no cuentan con un modo práctico para transmitir y recibir simultáneamente).

B Procedimiento para el modelado de flujos IP multimedia

1. ***Estudio teórico y planificación de la prueba.***
   (a) Recopilación de información de fuentes primarias y secundarias.
   (b) Análisis de la información.
   (c) Reproducción de ejemplos, simulaciones y similares.
   (d) Selección de aplicaciones a modelar.
   (e) Selección de equipamiento a utilizar.
   (f) Selección de herramientas para la obtención de datos.
   (g) Selección de herramientas para el análisis de resultados.
   (h) Asignación de tareas a grupos de trabajo.
   (i) Elaboración de un cronograma de actividades.
2. ***Acondicionamiento del entorno de pruebas.***
   (a) Proveer un entorno de pruebas libre de ruido e interferencia.
   (b) Realizar y comprobar conexiones físicas de red.
   (c) Puesta en marcha de equipos.
   (d) Configuración de parámetros básicas de red.
   (e) Comprobación de enlaces de red.
   (f) Configuración avanzada de cada elemento de red.
   (g) Monitorizar procesos activos y recursos.
   (h) Eliminación de procesos innecesarios en el sistema.
3. ***Obtención de resultados.***
   (a) Lanzar aplicaciones a utilizar.
   (b) Configuración de la aplicaciones.
   (c) Configuración de herramientas para la obtención de resultados
   (d) Lanzar herramientas para la captura de datos.
   (e) Iniciar la transmisión.

(f) Mantener activa transmisión según tiempo planificado.

(g) Terminar transmisión.

(h) Terminar herramienta de obtención de resultados.

4. ***Análisis de resultados.***

(a) Verificar contenidos de resultados obtenidos.

(b) Construir herramientas o procedimientos que permitan adaptar los resultados hacia las herramientas de análisis.

(c) Exportar resultados con formatos compatibles con las aplicaciones de análisis.

(d) Aplicar herramientas de análisis.

(e) Verificar la correspondencia de los resultados obtenidos con los protocolos involucrados.

5. ***Presentación de resultados.***

(a) Selección de las herramientas de elaboración de documentos.

(b) Selección de las herramientas de elaboración de gráfico y figuras.

(c) Representar síntesis de resultados con herramientas para la producción de documentos científicos.

C Aplicaciones multimedia

C.1 Voz IP

El desarrollo de tecnologías de VoIP ha tenido una gran aceptación por parte de empresas que buscan una reducción de costes para sus comunicaciones de voz principalmente PYMES (Pequeñas y Medianas Empresas) [5] y [12]. VoIP permite la transmisión de voz por medio de una red IP, como Internet, consiste en la digitalización de la señales de voz por medio de un *codec*, por otro lado, VoIP hace uso de diversos tipos de técnicas para la señalización de la llamada, no habiendo un protocolo definido en este ámbito. Uno de los protocolo utilizados con este fin en SIP (*Session Initiation Protocol*), además, se encuentran implementaciones con H.323 o IAX (*Inter-Asterisk eXchange protocol*).

SIP es uno de los protocolo con mayor impacto en la implementación de ToIP (*Telephony over IP*) [5], [11], [13] y [12], dicho protocolo, se encarga de la señalización extremo a extremo de la comunicación, realiza los procedimientos necesarios para el establecimiento de la llamada, la modificación y la finalización de la comunicación. Por otro lado, para la transmisión de datos en tiempo real, VoIP hace uso del protocolo RTP (*Real-time Transport Protocol*), dicho protocolo se encarga del control de la transmisión en las sesiones de aplicaciones multimedia y utiliza como protocolo de transporte UDP.

El dispositivo Linksys SPA 3102 es una pasarela de voz sobre IP hacia una red telefónica convencional y viceversa, utiliza el protocolo SIP (*Session Initiation Protocol*) para la señalización de la comunicación, además, puede ser configurado, con relativa facilidad, por medio de un menú IVR (*Interative Voice Response*) o mediante un servidor web desde cualquier navegador.

El menú IVR permite la configuración básica del dispositivo, como lo es la asignación de direcciones IP a cada terminal. La configuración avanzada del *gateway* VoIP consiste en la asignación de los parámetros correspondientes para las funcionalidades de enrutador y las asociadas a voz, esta configuración se realiza por medio de un navegador web. Dentro de los aspectos más relevantes de la configuración destacan la utilización de SIP como protocolo de señalización, la desactivación de NAT (*Network Address*

Translation) y demás funcionalidades de enrutamiento, por otro lado, en la configuración de audio se hace uso del *codec* *G*729 [2] y en la paquetización se definen dos muestras por paquete, no se realiza supresión de silencio.

De los resultados se obtiene la figura 21, en la que se ha reconstruido la distribución de los encabezados de cada uno de los paquetes capturados.

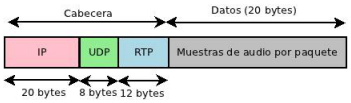


Figura 21: Paquete VoIP transmitido por las estaciones.

C.2 Cámaras de video sobre IP

Las cámaras IP han tenido un impacto importante en mecanismos de seguridad a nivel empresarial y residencial, este tipo de equipos permite emitir video utilizando técnicas de compresión de imagen a través de miles de kilómetros utilizando TCP/IP. Dentro de sus funciones se encuentran activación mediante movimiento o sensores, control remoto, gestión a través de HTTP, entre otros.

Para las pruebas realizadas en este trabajo se utiliza una cámara AXIS 2120 (un modelo diseñado para exteriores), esta cámara permite conectarse a redes *Ethernet* y *Fast Ethernet* con relativa facilidad, posee detector de movimientos, soporta protocolos como TCP/IP (*Transmision Control Protocol/Internet Protocol*), SMTP (*Simple Mail Transfer Protocol*), HTTP (*Hypertext Transfer Protocol*), entre otros. El formato de imagen es JPEG (*Joint Photographic Experts Group*) y soporta diferentes niveles de compresión.

En resumen, la cámara captura imágenes en formato JPEG y las transmite a razón de 25/30 tramas por segundo (PAL/NTSC) sobre una red con ancho de banda de 10 *Mbps* o 100 *Mbps* respectivamente.
La distribución de encabezados para un paquete de este tipo tiene la forma que se muestra en la figura 22. A diferencia del caso anterior, el transporte lo brinda TCP y la información está contenida en un paquete HTTP.

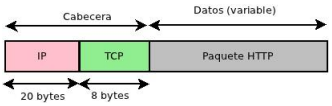


Figura 22: Paquete transmitido por la cámara.

C.3 Video streaming

El crecimiento a nivel mundial en el acceso a Internet por parte de los usuarios ha generado el desarrollo de diversas aplicaciones y nuevos modelos de negocio dentro de los que se encuentran la radio y la televisión por Internet (por mencionar algunos) [4] y [3]. Este tipo de servicios basa su funcionamiento en la transmisión *streaming*.

El *streaming* consiste en la distribución de audio o video por Internet, esta palabra hace referencia a una transmisión en forma continua, sin interrupciones y sin la necesidad de descargas previas. Para la implementación de este tipo de tráfico se ha escogido una aplicación ampliamente difundida y aceptada por los usuarios, como lo es VLC.

VLC media player [33] es un reproductor multimedia y *framework* multimedia libre y de código abierto desarrollado por el proyecto VideoLAN bajo la licencia GNU GPL. Es un programa multiplataforma con versiones disponibles para sistemas operativos como Microsoft Windows, GNU/Linux, Mac OS X, BeOS, BSD y eComStation, entre otros. El reproductor es capaz de reproducir muchos códecs y formatos de audio y video (dependiendo del sistema operativo en el que opere), además de capacidad de *streaming*.

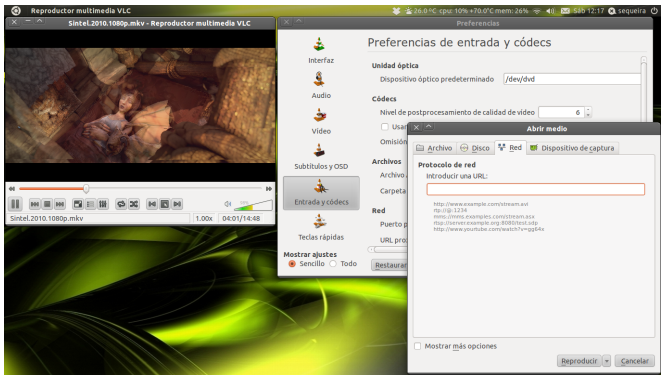


Figura 23: Reproductor de multimedia VLC.

Para proveer los servicios de video *streaming* se utiliza el reproductor de multimedia VLC, la captura obtenida es de aproximadamente 180 s de la película *"Sintel"* [24], una película de animación 3D y de libre distribución. Los resultados obtenidos muestran que los paquetes transmitidos tienen una pila de protocolos como se muestra en la figura 10, dichos resultados comprueban la configuración que se estableció para la aplicación, ya que destaca el uso del protocolo RTP y el uso de UDP para el transporte, el cual es de uso común para aplicaciones en tiempo real. En la configuración de la aplicación de *streaming* se ha desactivado la transcodificación ya que consume mucho procesador, por lo se reproduce en el receptor en el mismo formato del archivo original (MP4), así el procesador no se satura y la calidad en la recepción es muy buena, además, se establece el uso de RTP y MPEG-TS (MPEG Transport Stream).

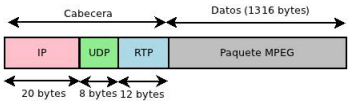


Figura 24: Paquete transmitido por el servidor.

C.4 Análisis de calidad en entornos reales en modo ad-hoc

Los sistemas 802.11 [34] poseen un protocolo de acceso al medio compartido denominado CSMA/CA (*Carrier Sense Multiple Access Collision Avoidance*), el cual define la manera en que los terminales comparten el medio. Cuando una estación quiere enviar una trama, verifica el canal en el que opera para detectar si existe una transmisión en alguna otra estación. Cuando el canal queda libre, la estación no transmite inmediatamente como se puede observar en la figura 25, sino, que debe seguir esperando el canal para asegurarse de que está libre durante un período de tiempo DIFS (*DCF InterFrame Space*), en ese momento puede transmitir una trama de datos.

Cuando una estación recibe de manera correcta una trama, espera un tiempo SIFS (*Short Interframe Space*) y manda la confirmación de reconocimiento (ACK). El SIFS tiene una duración de (10 μs) mientras que el DIFS de (52 μs).

Es posible que terminado el DIFS, dos o mas estaciones quieran transmitir al mismo tiempo, lo que provocaría colisiones en el caso de haber varias estaciones esperando transmitir. Por esto, se utilizan los *slots* de contención (con una duración de $20\,\mu s$ cada uno) o también denominado *backoff*, de tal manera, que cada estación que quiere transmitir calcula en forma aleatoria un valor de *slots* entre 0 y 31 y espera ese tiempo antes de transmitir.

Si aún así, se producen colisiones porque dos o mas estaciones han escogido el mismo valor, el procedimiento se repite escogiendo un valor entre 0 y 63, si continúa habiendo colisiones se va aumentando la cantidad de *slots* hasta un máximo de 1023. En el primer de los casos, el tiempo de *backoff* medio sin colisiones es de $15,5$ *slots* lo que corresponde a $310\,\mu s$.

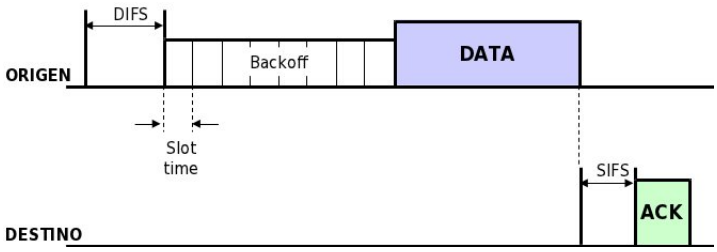


Figura 25: Relación temporal de una trama CSMA/CA.

Con las relaciones de tiempos de la figura 25 y conociendo la duración de transmisión de los datos (según el tipo de flujo IP multimedia, el cual tuvo que ser modelado previamente) y el ACK ($14\,bytes$) es posible determinar la cantidad de flujos que se pueden establecer en un enlace, con un ancho de banda determinado, en función de un determinado servicio. Lo primero que debe realizarse es determinar la longitud del paquete como se muestra en la ecuación 4, nótese que se ha agregado lo referente a la capa MAC de 802.11; después se calcula el tiempo que un paquete debe esperar para poder ser transmitido incluyendo el ACK como se muestra en la ecuación 5.

$$l = 10 \times m + RTP + UDP + IP + MAC_{wifi} \tag{4}$$

$$t_{notx} = (20\mu s \times slots) + DIFS + SIFS + 14 \times \left(\frac{8}{BW}\right) \tag{5}$$

Ahora, se podría obtener la cantidad de conexiones posibles de voz como la razón del tiempo que toman las m muestras con respecto al tiempo total desde que se inicia la transmisión de un paquete hasta el inicio del próximo paquete como aparece en la ecuación 6a (nótese que se utiliza $G729$ lo que corresponde al modelo obtenido en la sección 4.2.1), sin embargo, el valor obtenido en dicha ecuación corresponde a conexiones en un solo sentido; si se desea obtener para una comunicación *full duplex* se puede hacer uso de la ecuación 6b.

$$conexiones_{ow} = \frac{10ms \times m}{t_{notx} + preambulo + \frac{l}{BW}} \tag{6a}$$

$$conexiones_{rtt} = \frac{1}{2} \times conexiones_{ow} \tag{6b}$$

Para el caso de otro tipo de flujo IP, como por ejemplo, el de vídeo, se puede modificar la ecuación 6a de la siguiente forma:

$$conexiones_{ow} = \frac{tiempomediorafaga}{t_{notx} + preambulo + \frac{l}{BW}} \tag{7}$$

Es conveniente automatizar procesos repetitivos de cálculo con la finalidad de ahorrar tiempo en el análisis. En el apéndice D se muestra el *script* D.3 donde se realiza el cálculo de la cantidad de conexiones que se pueden establecer para un ancho de banda determinado en modo *one way* y *round trip*.

En el *script* D.3, así como en las ecuaciones anteriores, se puede apreciar que ciertas variables destacan en dicho cálculo, algunas de ellas son el ancho de banda del canal, la cantidad de muestras por paquete, el preámbulo y la cantidad de *slots* del *backoff*, este último tiene un efecto significativo debido a su naturaleza aleatoria, ya que si la cantidad de *slots* aumenta causa una reducción en la cantidad de flujos IP que se pueden establecer en un canal determinado para un mismo ancho de banda.

D *Script*

D.1 *Script* para el análisis de modelos para flujos de voz y video

```
%% Analisis de datos de trafico

%% Cargar archivos
clear all
voz=csvread(voz.csv);
camara_4=csvread(camara_4.csv);
camara_13=csvread(camara_13.csv);
camara_16=csvread(camara_16.csv);
camara_50=csvread(camara_50.csv);
video_streaming=csvread(video_streaming.csv);
umbral_voz=0;
umbral_camara_4=0.032;
umbral_camara_13=0.04;
umbral_camara_16=0.1;
umbral_camara_50=0.1;

%% Definir archivo
datos=camara_50;
umbral=umbral_camara_50;

%% Separar datos enviados y recibidos
a=1;
b=1;
for i=1:length(datos)
    if ((datos(i,3)==0) && (datos(i,4)==1))
        t_enviado(a)=datos(i,2);
        a=a+1;
    else
        t_recibido(b)=datos(i,2);
        b=b+1;
    end
end

%% Tiempos entre paquetes, medias y desviacion
t_paquete_enviado=zeros(size(t_enviado));
t_paquete_recibido=zeros(size(t_recibido));
for i=2: length(t_paquete_enviado)
    t_paquete_enviado(i)=t_enviado(i)-t_enviado(i-1);
end
```

```
40 for i=2: length(t_paquete_recibido)
41     t_paquete_recibido(i)=t_recibido(i)-t_recibido(i-1);
42 end
43 media_paquetes_enviados=mean(t_paquete_enviado);
44 media_paquetes_recibidos=mean(t_paquete_recibido);
45 desviacion_std_paquetes_enviados=std(t_paquete_enviado);
46 desviacion_std_paquetes_recibidos=std(t_paquete_recibido);
47
48 %% Tiempos entre rafagas, media y desviacion
49 c=1;
50 d=1;
51 for i=1:length(t_paquete_enviado)
52     if t_paquete_enviado(i)>umbral
53         t_rafaga_enviado(c)=t_paquete_enviado(i);
54         c=c+1;
55     end
56 end
57 for i=1:length(t_paquete_recibido)
58     if t_paquete_recibido(i)>umbral
59         t_rafaga_recibido(d)=t_paquete_recibido(i);
60         d=d+1;
61     end
62 end
63 media_rafaga_enviado=mean(t_rafaga_enviado);
64 media_rafaga_recibido=mean(t_rafaga_recibido);
65 desviacion_sdt_rafaga_enviado=std(t_rafaga_enviado);
66 desviacion_std_rafaga_recibido=std(t_rafaga_recibido);
67
68 %% Tiempo en el que se envio cada rafaga
69 e=1;
70 f=1;
71 for i=2:length(datos)
72     if ((datos(i,3)==0) && (datos(i,4)==1) &&
73             ((datos(i,2)-datos(i-1,2))>umbral))
74         tiempo_rafaga_enviado(e)=datos(i,2);
75         e=e+1;
76     else
77         if ((datos(i,3)==1) && (datos(i,4)==0) &&
78                 ((datos(i,2)-datos(i-1,2))>umbral))
79             tiempo_rafaga_recibido(f)=datos(i,2);
80             f=f+1;
81         end
82     end
83 end
84
```

```
85  %% Resumen estadistico
86  media_paquetes_enviados
87  desviacion_std_paquetes_enviados
88  media_paquetes_recibidos
89  desviacion_std_paquetes_recibidos
90  media_rafaga_enviado
91  desviacion_sdt_rafaga_enviado
92  media_rafaga_recibido
93  desviacion_std_rafaga_recibido
94  probabilidad_paquetes_enviados=length(t_enviado)/length(datos)
95  probabilidad_paquetes_recibidos=length(t_recibido)/length(datos)
96  probabilidad_rafagas_enviadas=
97          length(t_rafaga_enviado)/length(t_enviado)
98  probabilidad_rafagas_recibidas=
99          length(t_rafaga_recibido)/length(t_recibido)
100
101 %% Figuras
102
103 %Voz
104 figure(1);plot(t_enviado,t_paquete_enviado,r,
105         t_recibido,t_paquete_recibido,b)
106 figure(1);xlabel(Tiempo de transmision (s))
107 figure(1);ylabel(Duracion del paquete (s))
108 figure(1);title(Variacion del tiempo de paquetes para el
109         trafico de voz)
110 figure(1);legend(Trafico enviado,Trafico recibido)
111 figure(2);hist(t_paquete_enviado)
112 figure(2);xlabel(Duracion del paquete (s))
113 figure(2);ylabel(Cantidad de paquetes)
114 figure(2);title(Distribucion de paquetes para el trafico de voz)
115 figure(2);legend(Trafico enviado)
116 figure(3);hist(t_paquete_recibido)
117 figure(3);xlabel(Duracion del paquete (s))
118 figure(3);ylabel(Cantidad de paquetes)
119 figure(3);title(Distribucion de paquetes para el trafico de voz)
120 figure(3);legend(Trafico recibido)
121
122 % Camara 50
123 figure(1);subplot(2,1,1);plot(t_enviado,t_paquete_enviado,r,
124         t_recibido,t_paquete_recibido,b)
125 figure(1);subplot(2,1,1);xlabel(Tiempo de transmision (s))
126 figure(1);subplot(2,1,1);ylabel(Duracion del paquete (s))
127 figure(1);subplot(2,1,1);title(Variacion de la duracion de
128         paquetes)
129 figure(1);subplot(2,1,1);legend(Trafico enviado,Trafico recibido)
```

```
130 figure(1);subplot(2,1,2);plot(tiempo_rafaga_enviado,
131         t_rafaga_enviado)
132 figure(1);subplot(2,1,2);xlabel(Tiempo de transmision (s))
133 figure(1);subplot(2,1,2);ylabel(Duracion de la rafaga (s))
134 figure(1);subplot(2,1,2);title(Variacion de la duracion de
135         rafagas)
136 figure(1);subplot(2,1,2);legend(Trafico enviado)
137 figure(2);subplot(2,1,1);hist(t_paquete_enviado)
138 figure(2);subplot(2,1,1);xlabel(Duracion del paquete (s))
139 figure(2);subplot(2,1,1);ylabel(Cantidad de paquetes)
140 figure(2);subplot(2,1,1);title(Distribucion de paquetes enviados)
141 figure(2);subplot(2,1,1);legend(Trafico enviado)
142 figure(2);subplot(2,1,2);hist(t_paquete_recibido)
143 figure(2);subplot(2,1,2);xlabel(Duracion del paquete (s))
144 figure(2);subplot(2,1,2);ylabel(Cantidad de paquetes)
145 figure(2);subplot(2,1,2);title(Distribucion de paquetes recibidos)
146 figure(2);subplot(2,1,2);legend(Trafico recibido)
147
148 % Camara 16
149 figure(1);bar(t_enviado,t_paquete_enviado,r,
150         t_recibido,t_paquete_recibido,b)
151 figure(1);xlabel(Tiempo de transmision (s))
152 figure(1);ylabel(Duracion del paquete (s))
153 figure(1);title(Variacion del tiempo de paquetes para el
154         trafico de camara_16)
155 figure(1);legend(Trafico enviado,Trafico recibido)
156 figure(2);plot(tiempo_rafaga_enviado,t_rafaga_enviado)
157 figure(2);xlabel(Tiempo de transmision (s))
158 figure(2);ylabel(Duracion de la rafaga (s))
159 figure(2);title(Variacion del tiempo rafagas para el trafico
160         enviado de camara_16)
161 figure(2);legend(Trafico enviado)
162 figure(3);hist(t_paquete_enviado)
163 figure(3);xlabel(Duracion del paquete (s))
164 figure(3);ylabel(Cantidad de paquetes)
165 figure(3);title(Distribucion de paquetes para el trafico de
166         camara_16)
167 figure(3);legend(Trafico enviado)
168 figure(4);hist(t_paquete_recibido)
169 figure(4);xlabel(Duracion del paquete (s))
170 figure(4);ylabel(Cantidad de paquetes)
171 figure(4);title(Distribucion de paquetes para el trafico
172         camara_16)
173 figure(4);legend(Trafico enviado)
174
```

```
175 % Camara 13
176 figure(1);plot(t_enviado,t_paquete_enviado,r,
177         t_recibido,t_paquete_recibido,b)
178 figure(1);xlabel(Tiempo de transmision (s))
179 figure(1);ylabel(Duracion del paquete (s))
180 figure(1);title(Variacion del tiempo de paquetes para el trafico
181         de camara_13)
182 figure(1);legend(Trafico enviado,Trafico recibido)
183 figure(2);plot(tiempo_rafaga_enviado,t_rafaga_enviado)
184 figure(2);xlabel(Tiempo de transmision (s))
185 figure(2);ylabel(Duracion de la rafaga (s))
186 figure(2);title(Variacion del tiempo rafagas para el trafico
187         enviado de camara_13)
188 figure(2);legend(Trafico enviado)
189 figure(3);hist(t_paquete_enviado)
190 figure(3);xlabel(Duracion del paquetes para el trafico de
191         camara_13)
192 figure(3);ylabel(Cantidad de paquetes)
193 figure(3);title(Distribucion de paquetes para el trafico de
194         camara_13)
195 figure(3);legend(Trafico enviado)
196 figure(4);hist(t_paquete_recibido)
197 figure(4);xlabel(Duracion del paquete (s))
198 figure(4);ylabel(Cantidad de paquetes)
199 figure(4);title(Distribucion de paquetes para el trafico
200         camara_13)
201 figure(4);legend(Trafico recibido)
202
203 % Camara 4
204 figure(1);plot(t_enviado,t_paquete_enviado,r,
205         t_recibido,t_paquete_recibido,b)
206 figure(1);xlabel(Tiempo de transmision (s))
207 figure(1);ylabel(Duracion del paquete (s))
208 figure(1);title(Variacion del tiempo de paquetes para el trafico
209         de camara_4)
210 figure(1);legend(Trafico enviado,Trafico recibido)
211 figure(2);plot(tiempo_rafaga_enviado,t_rafaga_enviado)
212 figure(2);xlabel(Tiempo de transmision (s))
213 figure(2);ylabel(Duracion de la rafaga (s))
214 figure(2);title(Variacion del tiempo rafagas para el trafico
215         enviado de camara_4)
216 figure(2);legend(Trafico enviado)
217 figure(3);hist(t_paquete_enviado)
218 figure(3);xlabel(Duracion del paquete (s))
219 figure(3);ylabel(Cantidad de paquetes)
```

```
220 figure(3);title(Distribucion de paquetes para el trafico de
221         camara_4)
222 figure(3);legend(Trafico enviado)
223 figure(4);hist(t_paquete_recibido)
224 figure(4);xlabel(Duracion del paquete (s))
225 figure(4);ylabel(Cantidad de paquetes)
226 figure(4);title(Distribucion de paquetes para el trafico
227         camara_4)
228 figure(4);legend(Trafico recibido)
```

D.2 *Script* para el análisis de modelos para flujos *streaming*

```
1  %% Analisis de datos de trafico
2
3  clear all
4
5  %% Datos
6  voz=csvread(voz.csv);
7  camara_4=csvread(camara_4.csv);
8  camara_13=csvread(camara_13.csv);
9  camara_16=csvread(camara_16.csv);
10 camara_50=csvread(camara_50.csv);
11 video_streaming=csvread(streaming_3.csv);
12 umbral_voz=0;
13 umbral_camara_4=0.035;
14 umbral_camara_13=0.035;
15 umbral_camara_16=0.1;
16 umbral_camara_50=0.1;
17 umbral_streaming=20e-5;
18
19 %% Definir archivo
20 datos=video_streaming;
21 umbral=umbral_streaming;
22
23 %% Para streaming
24 if datos==video_streaming
25     a=1;
26     for i=1:length(datos)
27         if ((datos(i,3)==0) && (datos(i,4)==1))
28             t_enviado(a)=datos(i,2);
29             a=a+1;
30         end
```

```
31      end
32      t_paquete_enviado=zeros(size(t_enviado));
33      for i=2: length(t_paquete_enviado)
34          t_paquete_enviado(i)=t_enviado(i)-t_enviado(i-1);
35      end
36      c=1;
37      for i=1:length(t_paquete_enviado)
38          if t_paquete_enviado(i)>umbral
39              t_rafaga_enviado(c)=t_paquete_enviado(i);
40              c=c+1;
41          end
42      end
43      e=1;
44      for i=2:length(datos)
45          if ((datos(i,3)==0) && (datos(i,4)==1) && ((datos(i,2)-
46              datos(i-1,2))>umbral))
47              tiempo_rafaga_enviado(e)=datos(i,2);
48              e=e+1;
49          end
50      end
51      media_paquetes_enviados=mean(t_paquete_enviado)
52      desviacion_std_paquetes_enviados=std(t_paquete_enviado)
53      media_rafaga_enviado=mean(t_rafaga_enviado)
54      desviacion_sdt_rafaga_enviado=std(t_rafaga_enviado)
55      probabilidad_paquetes_enviados=length(t_enviado)/
56          length(datos)
57      probabilidad_rafagas_enviadas=length(t_rafaga_enviado)/
58          length(t_enviado)
59  end
60
61  %% Graficos
62
63  % subplot(2,2,[1 2]);plot(t_enviado,t_paquete_enviado)
64  % subplot(2,2,[1 2]);xlabel(Tiempo de transmision (s))
65  % subplot(2,2,[1 2]);ylabel(Duracion del paquete (s))
66  % subplot(2,2,[1 2]);title(Variacion del tiempo de paquetes para
67        el trafico streaming)
68  % subplot(2,2,[1 2]);legend(Trafico enviado)
69  % subplot(2,2,1);plot(t_enviado,t_paquete_enviado)
70  % subplot(2,2,1);xlabel(Tiempo de transmision (s))
71  % subplot(2,2,1);ylabel(Duracion del paquete (s))
72  % subplot(2,2,1);title(Variacion del tiempo de paquetes para
73        el trafico streaming)
74  % subplot(2,2,1);legend(Trafico enviado)
75  figure(5);subplot(2,2,[1 2]);scatter(tiempo_rafaga_enviado,
```

```
76      t_rafaga_enviado)
77  figure(5);subplot(2,2,[1 2]);xlabel(Tiempo de transmision (s))
78  figure(5);subplot(2,2,[1 2]);ylabel(Duracion de la rafaga (s))
79  figure(5);subplot(2,2,[1 2]);title(Variacion del tiempo rafagas
80      para el trafico enviado de streaming)
81  figure(5);subplot(2,2,[1 2]);legend(Trafico enviado)
82  % subplot(2,2,3);hist(t_paquete_enviado)
83  % subplot(2,2,3);xlabel(Duracion del paquete (s))
84  % subplot(2,2,3);ylabel(Cantidad de paquetes)
85  % subplot(2,2,3);title(Distribusion de paquetes para el
86      trafico streaming)
87  % subplot(2,2,3);legend(Trafico enviado)
88  figure(5);subplot(2,2,[3 4]);hist(t_rafaga_enviado)
89  figure(5);subplot(2,2,[3 4]);xlabel(Duracion de la rafaga (s))
90  figure(5);subplot(2,2,[3 4]);ylabel(Cantidad de rafagas)
91  figure(5);subplot(2,2,[3 4]);title(Distribusion de rafagas para
92      el trafico streaming)
93  figure(5);subplot(2,2,[3 4]);legend(Trafico enviado)
94
95  figure(1);plot(t_enviado,t_paquete_enviado)
96  figure(1);xlabel(Tiempo de transmision (s))
97  figure(1);ylabel(Duracion del paquete (s))
98  figure(1);title(Variacion del tiempo de paquetes para el trafico
99      streaming)
100 figure(1);legend(Trafico enviado)
101 % figure(2);scatter(tiempo_rafaga_enviado,t_rafaga_enviado)
102 % figure(2);xlabel(Tiempo de transmision (s))
103 % figure(2);ylabel(Duracion de la rafaga (s))
104 % figure(2);title(Variacion del tiempo rafagas para el trafico
105     enviado de streaming)
106 % figure(2);legend(Trafico enviado)
107 % figure(3);hist(t_paquete_enviado)
108 % figure(3);xlabel(Duracion del paquete (s))
109 % figure(3);ylabel(Cantidad de paquetes)
110 % figure(3);title(Distribusion de paquetes para el trafico
111     streaming)
112 % figure(3);legend(Trafico enviado)
113 % figure(4);hist(t_rafaga_enviado)
114 % figure(4);xlabel(Duracion del paquete (s))
115 % figure(4);ylabel(Cantidad de paquetes)
116 % figure(4);title(Distribusion de rafagas para el trafico
117     streaming)
118 % figure(4);legend(Trafico enviado)
```

D.3 *Script* para el cálculo de cantidad de conexiones

```
%% Calculo de cantidad de conversaciones de ToIP que se pueden
%% establecer en un determinado ancho de banda

%% Variables, modificar segun sea  el caso
clear all;
difs=52e-6;
sifs=10e-6;
slot=1; % Backoff media 15,5 slots
cp_media=20e-6*slot;
preamb=96e-6; % Corto, 2Mbps
%preamb=192e-6; % Largo, 1Mbps
bw_max=54e6;
g729=10e-3;
m=2;
ip=20;
udp=8;
rtp=12;
mac_wifi=34;

%% Calculos
for i=1:bw_max
    bw(i)=i;
end
for i=1:length(bw)
  % Longitud del paquete en bits
    l=(m*10+ip+udp+rtp+mac_wifi)*8;
    t_ack(i)=(14*(8/bw(i)))+preamb;
    % Tiempo de ACK medio sin coliciones
    t_ack_media(i)=cp_media+difs+sifs+t_ack(i);
  % Numero de conexiones en un sentido
    num_ow(i)=m*g729/(t_ack_media(i)+preamb+(l/bw(i)));
  % Numero de conexiones bidireccionales
    num_rt(i)=m*g729/(2*(t_ack_media(i)+preamb+(l/bw(i))));
end
figure(1);plot(bw, num_rt, b, bw,num_ow, r)
figure(1);xlabel(Ancho de banda (Hz))
figure(1);ylabel(Numero de conexiones)
figure(1);title(Cantidad de conexiones de VoIP para un enlace
        WiFi en modo ad-hoc en funcion del ancho de
        banda del canal)
figure(1);legend(Round trip, One way)
```

D.4 *Script* para generar archivo del tráfico de vídeo con una compresión de $4\,Kbytes$

```
%% Generar tamanos y tiempos para 4K

clear all;
close all;

total_paquetes=10000;

%% Creando datos
for i=1:3:total_paquetes
    datos(i,1)=40000; % microsegundos
    datos(i,2)=1472;
end

for i=2:3:total_paquetes
    datos(i,1)=2000; % microsegundos
    datos(i,2)=1472;
end

for i=3:3:total_paquetes
    datos(i,1)=4000; % microsegundos
    datos(i,2)=1472;
end

dlmwrite(trafico_enviar_4k.txt,datos,newline,pc,precision,%.0f);
```

D.5 *Script* para generar archivo del tráfico de vídeo con una compresión de $50\,Kbytes$

```
%% Generador de archivo de tamanos y tiempos
%% Datos

clear all
close all

%format bank;
paquetes_rafaga=25;
total_paquetes=10000;
rafagas=[80 23; 120 99; 160 171; 200 30];
porcentaje_rafagas=[7.12 37.77 90.71 100];
```

```
12 tamano_paquete=1472;
13 porcentaje_paquetes_40=1.32;
14 cont1=1;
15 cont2=1;
16 cont3=1;
17 cont4=1;
18 cont5=1;
19 cont_raf_80=1;
20 cont_raf_120=1;
21 cont_raf_160=1;
22 cont_raf_200=1;
23
24 %% Construir tamanos
25 for i=1:total_paquetes
26     if cont1==paquetes_rafaga
27         tamanos_tiempos(i,1)=floor(random(unif,700,1300));
28         cont1=1;
29     else
30         tamanos_tiempos(i,1)=tamano_paquete;
31         cont1=cont1+1;
32     end
33 end
34
35 %% Construir tiempos
36 tamanos_tiempos(:,2)=0;
37 for j=1:total_paquetes
38     if tamanos_tiempos(j,2)==0
39         for i=1:paquetes_rafaga:total_paquetes
40             %tipo_rafaga=floor(random(unif,0,100));
41             tipo_rafaga=unifrnd(0,100);
42             if tamanos_tiempos(i,2)==0
43                 if tipo_rafaga<=porcentaje_rafagas(1)
44                     misc=i;
45                     while cont2<=25
46                         tamanos_tiempos(misc,2)=
47                             random(unif,1.75e-4,2.75e-4);
48                         tamanos_tiempos(misc,3)=80;
49                         cont2=cont2+1;
50                         misc=misc+1;
51                     end
52                     cont2=1;
53                 else
54                     if tipo_rafaga<=porcentaje_rafagas(2)
55                         misc=i;
56                         while cont3<=25
```

```
57                                  tamanos_tiempos(misc,2)=
58                                      random(unif,1.75e-4,2.75e-4);
59                                  tamanos_tiempos(misc,3)=120;
60                                  cont3=cont3+1;
61                                  misc=misc+1;
62                              end
63                              cont3=1;
64                          else
65                              if tipo_rafaga<=porcentaje_rafagas(3)
66                                  misc=i;
67                                  while cont4<=25
68                                      tamanos_tiempos(misc,2)=
69                                          random(unif,1.75e-4,2.75e-4);
70                                      tamanos_tiempos(misc,3)=160;
71                                      cont4=cont4+1;
72                                      misc=misc+1;
73                                  end
74                                  cont4=1;
75                              else
76                                  if tipo_rafaga<=porcentaje_rafagas(4)
77                                      misc=i;
78                                      while cont5<=25
79                                          tamanos_tiempos(misc,2)=
80                                            random(unif,1.75e-4,2.75e-4);
81                                          tamanos_tiempos(misc,3)=200;
82                                          cont5=cont5+1;
83                                          misc=misc+1;
84                                      end
85                                      cont5;
86                                  end
87                              end
88                          end
89                      end
90                  end
91              end
92          end
93      end
94
95      %% Calculando el acumulado de tiempos
96      for i=1:total_paquetes
97          if i==1
98              tamanos_tiempos(i,4)=tamanos_tiempos(i,2);
99          else
100             tamanos_tiempos(i,4)=tamanos_tiempos(i,2)+
101                 tamanos_tiempos(i-1,4);
```

```
102     end
103 end
104
105 %% Calculando tiempos entre rafagas
106 tamanos_tiempos(:,3)=tamanos_tiempos(:,3)*1e-3;
107 for i=1:total_paquetes-1
108     if tamanos_tiempos(i,1)~=tamano_paquete
109         tamanos_tiempos(i+1,2)=tamanos_tiempos(i+1,2)+
110             tamanos_tiempos(i,3);
111     end
112 end
113
114 datos(:,2)=tamanos_tiempos(:,1);
115 datos(:,1)=floor(tamanos_tiempos(:,2)*1e6);
116 dlmwrite(trafico_enviar.txt,datos,newline,pc,precision,%.0f);
117
118 %% Para comparar los resultados simulados y los reales
119 figure(1);bar(rafagas(:,1),rafagas(:,2));
120 figure(2);hist(tamanos_tiempos(1:total_paquetes,3));
```

D.6 *Script* para el cálculo del tamaño del *buffer*

```
1  %% Determinar caracteristicas del buffer
2
3  clear all
4
5  %% Entrada de capturas
6  % Interfaz en minipc4 Tx
7  datos_em1_54=csvread(54Mbps/datos_captura_buffer_em1_1300.csv);
8   % Interfaz en minipc3 Rx (supone el tiempo mas largo)
9  datos_p5p1_54=csvread(54Mbps/datos_captura_buffer_p5p1_1300.csv);
10  % Interfaz en minipc4 Tx
11 datos_em1_24=csvread(24Mbps/datos_captura_buffer_em1_1300.csv);
12 % Interfaz en minipc3 Rx (supone el tiempo mas largo)
13 datos_p5p1_24=csvread(24Mbps/datos_captura_buffer_p5p1_1300.csv);
14 % Interfaz en minipc4 Tx
15 datos_em1_11=csvread(11Mbps/datos_captura_buffer_em1_1300.csv);
16 % Interfaz en minipc3 Rx (supone el tiempo mas largo)
17 datos_p5p1_11=csvread(11Mbps/datos_captura_buffer_p5p1_1300.csv);
18 % Interfaz en minipc4 Tx
19 datos_em1_5=csvread(5.5Mbps/datos_captura_buffer_em1_1300.csv);
20 % Interfaz en minipc3 Rx (supone el tiempo mas largo)
21 datos_p5p1_5=csvread(5.5Mbps/datos_captura_buffer_p5p1_1300.csv);
22 % Interfaz en minipc4 Tx
```

```
23 datos_em1_2=csvread(2Mbps/datos_captura_buffer_em1_1300.csv);
24 % Interfaz en minipc3 Rx (supone el tiempo mas largo)
25 datos_p5p1_2=csvread(2Mbps/datos_captura_buffer_p5p1_1300.csv);
26 % Interfaz en minipc4 Tx
27 datos_em1_1=csvread(1Mbps/datos_captura_buffer_em1_1300.csv);
28 % Interfaz en minipc3 Rx (supone el tiempo mas largo)
29 datos_p5p1_1=csvread(1Mbps/datos_captura_buffer_p5p1_1300.csv);
30
31 %% Analisis de enlace de 54Mbps
32 % Crear matriz de datos para los paquetes recibidos
33 paquetes_iguales=0;
34 for i=1:length(datos_p5p1_54)
35     for j=1:length(datos_em1_54)
36         if (datos_em1_54(j,1)==datos_p5p1_54(i,1)) &&
37             (datos_em1_54(j,2)==datos_p5p1_54(i,2))
38             paquetes_iguales=paquetes_iguales+1;
39             % Identificador de conexion
40             datos_54(paquetes_iguales,1)=datos_em1_54(j,1);
41             % Identificador de paquete
42             datos_54(paquetes_iguales,2)=datos_em1_54(j,2);
43             % Tamano de paquete
44             datos_54(paquetes_iguales,3)=datos_em1_54(j,5);
45             % Tiempo de captura en Tx
46             datos_54(paquetes_iguales,4)=datos_em1_54(j,4)-
47                 datos_em1_54(1,4);
48             % Tiempo de captura en Rx
49             datos_54(paquetes_iguales,5)=datos_p5p1_54(i,4)-
50                 datos_p5p1_54(1,4);
51             % Retardo del paquete
52             datos_54(paquetes_iguales,6)=datos_p5p1_54(i,4)-
53                 datos_em1_54(j,4);
54             datos_54(paquetes_iguales,7)=0;
55         end
56     end
57 end
58
59 % Tamano del buffer
60 for i=1:length(datos_54)
61     seguir=1;
62     for j=1:length(datos_54)
63         if datos_54(i,4)<datos_54(j,5) && seguir==1
64             datos_54(i,7)=abs(i-j);
65             seguir=0;
66         end
67     end
```

```
68  end
69
70  %% Analisis de enlace de 24Mbps
71  % Crear matriz de datos para los paquetes recibidos
72  paquetes_iguales=0;
73  for i=1:length(datos_p5p1_24)
74      for j=1:length(datos_em1_24)
75          if (datos_em1_24(j,1)==datos_p5p1_24(i,1)) &&
76              (datos_em1_24(j,2)==datos_p5p1_24(i,2))
77              paquetes_iguales=paquetes_iguales+1;
78              % Identificador de conexion
79              datos_24(paquetes_iguales,1)=datos_em1_24(j,1);
80              % Identificador de paquete
81              datos_24(paquetes_iguales,2)=datos_em1_24(j,2);
82              % Tamano de paquete
83              datos_24(paquetes_iguales,3)=datos_em1_24(j,5);
84              % Tiempo de captura en Tx
85              datos_24(paquetes_iguales,4)=datos_em1_24(j,4)-
86                  datos_em1_24(1,4);
87              % Tiempo de captura en Rx
88              datos_24(paquetes_iguales,5)=datos_p5p1_24(i,4)-
89                  datos_p5p1_24(1,4);
90              % Retardo del paquete
91              datos_24(paquetes_iguales,6)=datos_p5p1_24(i,4)-
92                  datos_em1_24(j,4);
93              datos_24(paquetes_iguales,7)=0;
94          end
95      end
96  end
97
98  % Tamano del buffer
99  for i=1:length(datos_24)
100     seguir=1;
101     for j=1:length(datos_24)
102         if datos_24(i,4)<datos_24(j,5) && seguir==1
103             datos_24(i,7)=abs(i-j);
104             seguir=0;
105         end
106     end
107 end
108
109 %% Analisis de enlace de 11Mbps
110 % Crear matriz de datos para los paquetes recibidos
111 paquetes_iguales=0;
112 for i=1:length(datos_p5p1_11)
```

```
113     for j=1:length(datos_em1_11)
114         if (datos_em1_11(j,1)==datos_p5p1_11(i,1)) &&
115             (datos_em1_11(j,2)==datos_p5p1_11(i,2))
116             paquetes_iguales=paquetes_iguales+1;
117             % Identificador de conexion
118             datos_11(paquetes_iguales,1)=datos_em1_11(j,1);
119             % Identificador de paquete
120             datos_11(paquetes_iguales,2)=datos_em1_11(j,2);
121             % Tamano de paquete
122             datos_11(paquetes_iguales,3)=datos_em1_11(j,5);
123             % Tiempo de captura en Tx
124             datos_11(paquetes_iguales,4)=datos_em1_11(j,4)-
125                 datos_em1_11(1,4);
126             % Tiempo de captura en Rx
127             datos_11(paquetes_iguales,5)=datos_p5p1_11(i,4)-
128                 datos_p5p1_11(1,4);
129             % Retardo del paquete
130             datos_11(paquetes_iguales,6)=datos_p5p1_11(i,4)-
131                 datos_em1_11(j,4);
132             datos_11(paquetes_iguales,7)=0;
133         end
134     end
135 end
136
137 % Tamano del buffer
138 for i=1:length(datos_11)
139     seguir=1;
140     for j=1:length(datos_11)
141         if datos_11(i,4)<datos_11(j,5) && seguir==1
142             datos_11(i,7)=abs(i-j);
143             seguir=0;
144         end
145     end
146 end
147
148 %% Analisis de enlace de 5.5Mbps
149 % Crear matriz de datos para los paquetes recibidos
150 paquetes_iguales=0;
151 for i=1:length(datos_p5p1_5)
152     for j=1:length(datos_em1_5)
153         if (datos_em1_5(j,1)==datos_p5p1_5(i,1)) &&
154             (datos_em1_5(j,2)==datos_p5p1_5(i,2))
155             paquetes_iguales=paquetes_iguales+1;
156             % Identificador de conexion
157             datos_5(paquetes_iguales,1)=datos_em1_5(j,1);
```

```
158             % Identificador de paquete
159             datos_5(paquetes_iguales,2)=datos_em1_5(j,2);
160             % Tamano de paquete
161             datos_5(paquetes_iguales,3)=datos_em1_5(j,5);
162             % Tiempo de captura en Tx
163             datos_5(paquetes_iguales,4)=datos_em1_5(j,4)-
164                 datos_em1_5(1,4);
165             % Tiempo de captura en Rx
166             datos_5(paquetes_iguales,5)=datos_p5p1_5(i,4)-
167                 datos_p5p1_5(1,4);
168             % Retardo del paquete
169             datos_5(paquetes_iguales,6)=datos_p5p1_5(i,4)-
170                 datos_em1_5(j,4);
171             datos_5(paquetes_iguales,7)=0;
172         end
173     end
174 end
175
176 % Tamano del buffer
177 for i=1:length(datos_5)
178     seguir=1;
179     for j=1:length(datos_5)
180         if datos_5(i,4)<datos_5(j,5) && seguir==1
181             datos_5(i,7)=abs(i-j);
182             seguir=0;
183         end
184     end
185 end
186
187 %% Analisis de enlace de 2Mbps
188 % Crear matriz de datos para los paquetes recibidos
189 paquetes_iguales=0;
190 for i=1:length(datos_p5p1_2)
191     for j=1:length(datos_em1_2)
192         if (datos_em1_2(j,1)==datos_p5p1_2(i,1)) &&
193             (datos_em1_2(j,2)==datos_p5p1_2(i,2))
194             paquetes_iguales=paquetes_iguales+1;
195             % Identificador de conexion
196             datos_2(paquetes_iguales,1)=datos_em1_2(j,1);
197             % Identificador de paquete
198             datos_2(paquetes_iguales,2)=datos_em1_2(j,2);
199             % Tamano de paquete
200             datos_2(paquetes_iguales,3)=datos_em1_2(j,5);
201             % Tiempo de captura en Tx
202             datos_2(paquetes_iguales,4)=datos_em1_2(j,4)-
```

```
203                 datos_em1_2(1,4);
204             % Tiempo de captura en Rx
205             datos_2(paquetes_iguales,5)=datos_p5p1_2(i,4)-
206                 datos_p5p1_2(1,4);
207             % Retardo del paquete
208             datos_2(paquetes_iguales,6)=datos_p5p1_2(i,4)-
209                 datos_em1_2(j,4);
210             datos_2(paquetes_iguales,7)=0;
211         end
212     end
213 end
214
215 % Tamano del buffer
216 for i=1:length(datos_2)
217     seguir=1;
218     for j=1:length(datos_2)
219         if datos_2(i,4)<datos_2(j,5) && seguir==1
220             datos_2(i,7)=abs(i-j);
221             seguir=0;
222         end
223     end
224 end
225
226 %% Analisis de enlace de 1Mbps
227 % Crear matriz de datos para los paquetes recibidos
228 paquetes_iguales=0;
229 for i=1:length(datos_p5p1_1)
230     for j=1:length(datos_em1_1)
231         if (datos_em1_1(j,1)==datos_p5p1_1(i,1)) &&
232             (datos_em1_1(j,2)==datos_p5p1_1(i,2))
233             paquetes_iguales=paquetes_iguales+1;
234             % Identificador de conexion
235             datos_1(paquetes_iguales,1)=datos_em1_1(j,1);
236             % Identificador de paquete
237             datos_1(paquetes_iguales,2)=datos_em1_1(j,2);
238             % Tamano de paquete
239             datos_1(paquetes_iguales,3)=datos_em1_1(j,5);
240             % Tiempo de captura en Tx
241             datos_1(paquetes_iguales,4)=datos_em1_1(j,4)-
242                 datos_em1_1(1,4);
243             % Tiempo de captura en Rx
244             datos_1(paquetes_iguales,5)=datos_p5p1_1(i,4)-
245                 datos_p5p1_1(1,4);
246             % Retardo del paquete
247             datos_1(paquetes_iguales,6)=datos_p5p1_1(i,4)-
```

```
248                 datos_em1_1(j,4);
249             datos_1(paquetes_iguales,7)=0;
250         end
251     end
252 end
253
254 % Tamano del buffer
255 for i=1:length(datos_1)
256     seguir=1;
257     for j=1:length(datos_1)
258         if datos_1(i,4)<datos_1(j,5) && seguir==1
259             datos_1(i,7)=abs(i-j);
260             seguir=0;
261         end
262     end
263 end
264
265 %% Graficas
266 figure(1);subplot(3,1,1);plot(datos_54(1:114,4),datos_54(1:114,7))
267 figure(1);subplot(3,1,1);xlabel(Referencia de tiempo del
268     procesador)
269 figure(1);subplot(3,1,1);ylabel(Ocupacion del buffer)
270 figure(1);subplot(3,1,1);title(Ocupacion del buffer de un
271     router de acceso)
272 figure(1);subplot(3,1,1);legend(54Mbps)
273
274 figure(1);subplot(3,1,2);plot(datos_24(1:114,4),datos_24(1:114,7))
275 figure(1);subplot(3,1,2);xlabel(Referencia de tiempo del
276     procesador)
277 figure(1);subplot(3,1,2);ylabel(Ocupacion del buffer)
278 figure(1);subplot(3,1,2);title(Ocupacion del buffer de un
279     router de acceso)
280 figure(1);subplot(3,1,2);legend(24Mbps)
281
282 figure(1);subplot(3,1,3);plot(datos_11(1:114,4),datos_11(1:114,7))
283 figure(1);subplot(3,1,3);xlabel(Referencia de tiempo del
284     procesador)
285 figure(1);subplot(3,1,3);ylabel(Ocupacion del buffer)
286 figure(1);subplot(3,1,3);title(Ocupacion del buffer de un
287     router de acceso)
288 figure(1);subplot(3,1,3);legend(11Mbps)
289
290 figure(2);subplot(3,1,1);plot(datos_5(1:114,4),datos_5(1:114,7))
291 figure(2);subplot(3,1,1);xlabel(Referencia de tiempo del
292     procesador)
```

```
293 figure(2);subplot(3,1,1);ylabel(Ocupacion del buffer)
294 figure(2);subplot(3,1,1);title(Ocupacion del buffer de un
295     router de acceso)
296 figure(2);subplot(3,1,1);legend(5.5Mbps)
297
298 figure(2);subplot(3,1,2);plot(datos_2(1:114,4),datos_2(1:114,7))
299 figure(2);subplot(3,1,2);xlabel(Referencia de tiempo del
300     procesador)
301 figure(2);subplot(3,1,2);ylabel(Ocupacion del buffer)
302 figure(2);subplot(3,1,2);title(Ocupacion del buffer de un
303     router de acceso)
304 figure(2);subplot(3,1,2);legend(2Mbps)
305
306 figure(2);subplot(3,1,3);plot(datos_1(1:114,4),datos_1(1:114,7))
307 figure(2);subplot(3,1,3);xlabel(Referencia de tiempo del
308     procesador)
309 figure(2);subplot(3,1,3);ylabel(Ocupacion del buffer)
310 figure(2);subplot(3,1,3);title(Ocupacion del buffer de un
311     router de acceso)
312 figure(2);subplot(3,1,3);legend(1Mbps)
313
314 figure(3);plot(datos_54(1:114,4),datos_54(1:114,7),
315   datos_24(1:114,4),datos_24(1:114,7),datos_11(1:114,4),
316   datos_11(1:114,7),datos_5(1:114,4),datos_5(1:114,7),
317   datos_2(1:114,4),datos_2(1:114,7),datos_1(1:114,4),
318   datos_1(1:114,7))
319 figure(3);xlabel(Referencia de tiempo del procesador (s))
320 figure(3);ylabel(Ocupacion del buffer)
321 figure(3);title(Ocupacion del buffer de un router de acceso)
322 figure(3);legend(54Mbps,24Mbps,11Mbps,5.5Mbps,2Mbps,1Mbps)
323
324 figure(4);plot(datos_54(:,4),datos_54(:,7),datos_24(:,4),
325   datos_24(:,7),datos_11(:,4),datos_11(:,7),datos_5(:,4),
326   datos_5(:,7),datos_2(:,4),datos_2(:,7),datos_1(:,4),
327   datos_1(:,7))
328 figure(4);xlabel(Referencia de tiempo del procesador (s))
329 figure(4);ylabel(Ocupacion del buffer)
330 figure(4);title(Ocupacion del buffer de un router de acceso)
331 figure(4);legend(54Mbps,24Mbps,11Mbps,5.5Mbps,2Mbps,1Mbps)
```