

ESCUELA TÉCNICA DE INGENIERÍA DE
TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE CARTAGENA



PROYECTO FIN DE CARRERA

ANÁLISIS DE LA CALIDAD EXPERIMENTADA EN
APLICACIONES DE VOZ SOBRE IP DE LIBRE
DISTRIBUCIÓN



AUTOR: Adoración Marín Soler
DIRECTOR: M^a Dolores Cano Baños
Cartagena, Septiembre 2008



Autor	Adoración Marín Soler
E-mail del Autor	dori_m_s@hotmail.com
Director	M ^a Dolores Cano Baños
E-mail del Director	Mdolores.cano@upct.es
Título del PFC	Análisis de la Calidad Experimentada en Aplicaciones de Voz sobre IP de Libre Distribución
Resumen	<p>En este proyecto el trabajo que se ha realizado ha consistido en determinar los parámetros de la red que determinan la calidad de servicio de las aplicaciones de VoIP y comparar los resultados obtenidos con las encuestas realizadas a un número de personas.</p> <p>Para ello, primeramente se ha realizado un estudio de las características y funcionamiento de los protocolos para VoIP. Seguidamente se eligieron 5 aplicaciones de VoIP y se realizaron capturas de su transferencia de paquetes. Con estos datos se ha podido calcular los parámetros de la red y se ha establecido de este modo la calidad de servicio ofrecida. El análisis de las aplicaciones se realizará mediante capturas con la herramienta wireshark, que serán analizadas posteriormente utilizando awk.</p> <p>Por otro lado para medir la QoE se han realizado encuestas de las mismas aplicaciones a un número elevado de personas.</p> <p>Con estos datos, hemos podido establecer una comparación de los resultados obtenidos en ambos estudios para realizar una posterior conclusión de los mismos.</p>
Titulación	I.T.T TELEMÁTICA
Intensificación	
Departamento	Departamento de Tecnologías de Información y Comunicaciones
Fecha de presentación	Septiembre 2008

ÍNDICE

Capítulo 1. Motivación del proyecto	3
1.1 Objetivos del proyecto	3
1.2 Descripción del proyecto	3
Capítulo 2. Aplicaciones de VoIP para la empresa	5
2.1 Introducción	5
2.2 Definición de VoIP	5
2.2.1 Ventajas de una red VoIP	6
2.2.2 Desventajas de una red VoIP	7
2.2.3 Comparación entre PSTN y VoIP	8
2.3 Consideraciones de diseño de una red de VoIP	9
2.3.1 Muestrear y cuantificar	9
2.3.2 Implementación	9
2.4 Mecanismos que garantizan la Calidad de Servicio	15
2.4.1 Codificadores y decodificadores	16
2.4.2 Clasificación de los paquetes	19
2.4.3 Mecanismos de cola	19
2.4.4 Evitar la congestión	21
2.4.5 Fragmentación e inserción	21
2.4.6 Desordenación de los paquetes	21
2.5 Protocolos de transporte	22
2.6 Protocolos de señalización	23
2.7 Equipos de una red VoIP	26
2.7.1 Componentes de VoIP y equipo necesario	26
2.7.2 Papel desempeñado por el FCC en VoIP	28
2.7.3 Ejemplo de una llamada en VoIP	28
2.8 Aplicaciones de VoIP	29
Capítulo 3. Análisis de QoS en aplicaciones de VoIP de libre distribución	35
3.1 Introducción	35
3.2 Parámetros usados para el análisis	35
3.2.1 Ancho de banda	36
3.2.2 Jitter	36
3.2.3 Tiempo entre paquetes	36
3.2.4 Paquetes perdidos	36
3.3 Descripción de las mediciones a tomar	37
3.4 Herramientas de análisis	37
3.4.1 Wireshark	37
3.4.2 Script en awk	38
3.5 Resultados y discusión	41
3.5.1 Tiempo medio entre paquetes	41
3.5.2 Varianza del tiempo entre paquetes	45
3.5.3 Ancho de banda medio	49

3.5.4	Varianza del ancho de banda	52
3.5.5	Jitter medio	53
3.5.6	Varianza del jitter	55
3.5.7	Paquetes perdidos	58
3.6	Conclusiones	61
Capítulo 4. Análisis de QoE en aplicaciones de libre distribución		63
4.1	Introducción	63
4.2	Definición de QoE	63
4.3	Medir la QoE	64
4.3.1	Medir la QoE de video	65
4.3.2	Medir la QoE de la voz	65
4.4	Entorno de las mediciones	68
4.4.1	Video	68
4.4.2	Voz	69
4.4.3	Otros datos de interés	70
4.5	Análisis de QoS de las conversaciones de los encuestados	70
4.6	Resultados	72
4.6.1	Resultados que espera el usuario de la aplicación	72
4.6.2	interfaz gráfica	73
4.6.3	Tiempo que tarda la aplicación en conectarse	74
4.6.4	Inestabilidad	75
4.6.5	Utilidad	76
4.6.6	Distorsión de la voz	77
4.6.7	Eco	78
4.6.8	Ruido	78
4.6.9	Pérdidas	79
4.6.10	Retardo	80
4.6.11	Puntuación MOS	81
4.6.12	Calidad de video	82
4.7	Comparativa QoS vs QoE	82
4.7.1	Trabajos previos	83
4.7.2	Efectos de los parámetros medidos	83
4.7.3	Comparación de resultados de QoS y QoE	84
Capítulo 5. Conclusiones		93
Referencias		95
Bibliografía		99

Capítulo 1

Motivación del proyecto

1.1 Objetivos del proyecto

En los últimos años, las aplicaciones de VoIP han experimentado un gran desarrollo en sus características. Las mayores ventajas que presentan son tan importantes como un coste prácticamente nulo, una gran facilidad de instalación y una disponibilidad permanente del servicio.

Debido a la mejora de estas características, principalmente al ahorro que en una pequeña empresa se puede producir debido a su implantación, son muchas las que se plantean su desarrollo. Sin embargo, el porcentaje de las que finalmente se deciden a su instalación es muy bajo. Las causas principales de esta reticencia son debidas a los malos resultados de calidad que en un pasado ofrecía VoIP y al escaso conocimiento que se posee de esta tecnología.

En este proyecto se pretende ilustrar la calidad de servicio que es ofrecida por este tipo de aplicaciones y su comparación con la calidad experimentada por un grupo de usuarios. De esta manera se estaría realizando una valoración tanto objetiva como subjetiva de sus características. Ambos parámetros aportan unos resultados finales necesarios para la realización de una valoración global de las aplicaciones de VoIP. De esta manera se solventarían las dudas que generan este tipo de aplicaciones antes de su implantación.

Para llevar a cabo esta tarea se ha dividido el proyecto en diversos capítulos que a continuación vamos a describir brevemente.

1.2 Descripción del proyecto

En el capítulo 2 se hace una descripción de VoIP. Se mencionan los parámetros de diseño necesarios para una transmisión exitosa, los protocolos que utiliza, codificadores, así como las ventajas e inconvenientes que la caracterizan frente a otras tecnologías. Para poder realzar las mediciones de QoS y QoE se eligen 5 aplicaciones de entre las que proporcionan mejores prestaciones, siendo Gizmo, Skype, Damaka, ooVoo y vBuzzer las elegidas. Finalmente, se describen las principales características de las aplicaciones seleccionadas.

En el capítulo 3 se analizan los diferentes parámetros de QoS que hemos medido para cada una de las aplicaciones de libre distribución seleccionadas en el capítulo 2. En concreto se medirá el tiempo entre paquetes, el retardo, el ancho de banda, el jitter y el porcentaje de paquetes perdidos. Todos estos parámetros se medirán en media excepto los dos primeros que además se medirán en varianza. La herramienta utilizada para las capturas del tráfico generado por las aplicaciones es Wireshark [WIR08]. Estos datos serán analizados mediante la herramienta awk [AWK08] con la que se ha creado un script que nos permite obtener los valores de los parámetros bajo estudio. Las mediciones se realizarán en tres tramos horarios, mañana, tarde y noche, que nos proporcionarán diversos valores dependiendo de la carga experimentada en la red.

En el capítulo 4 se analizan los parámetros de QoE que los usuarios han valorado de cada aplicación. Para ello se realizaron encuestas en las que los sujetos utilizaban las mismas herramientas de libre distribución que se han utilizado en el capítulo anterior para la medición de la QoS. Mediante el uso de la aplicación han valorado distintos parámetros. A partir de esos datos se han establecido gráficas que muestran los distintos resultados que muestra cada aplicación.

En el capítulo 5 se realiza la comparación entre QoS y QoE mediante las conclusiones obtenidas de cada estudio realizado. Se valora si los parámetros objetivos hallados mediante el cálculo de las capturas se corresponden con los valores subjetivos que los usuarios han puntuado. De esta manera se comprueba si existe una relación entre ambos que pueda determinar la elección de una aplicación u otra para un uso posterior.

Con los datos de los capítulos anteriores podemos determinar en este último las conclusiones finales que se han derivado de todo el estudio anteriormente descrito. Determinando la relación existente entre la QoE y la QoS podríamos decidir cuál de las aplicaciones bajo estudio es la que proporciona mejores resultados.

Capítulo 2

Aplicaciones de VoIP para la empresa

2.1 Introducción

En este capítulo vamos a hablar de VoIP (*Voice over Internet Protocol*; Voz sobre el Protocolo de Internet). Para ello analizaremos cuáles son las características apropiadas que debe tener una red para que la voz se pueda transmitir de forma correcta considerando sus protocolos de transporte, señalización y los codificadores necesarios para su transmisión. Además comentaremos las ventajas e inconvenientes que caracterizan a VoIP.

Terminaremos este capítulo mencionando los equipos necesarios para montar una red de voz sobre IP (*Internet Protocol*, Protocolo de Internet) [RFC791] y realizando un ejemplo de llamada y describiendo las aplicaciones de VoIP que hemos seleccionado para este trabajo.

2.2 Definición de VoIP

Voz sobre IP es una tecnología que permite realizar llamadas telefónicas sobre redes de ordenadores, como puede ser *Internet*. Voz sobre IP también es conocida como la telefonía de *Internet*. La telefonía tradicional consiste en transportar la señal analógica sobre cable de cobre, pero la tecnología de VoIP convierte la voz analógica en paquetes de datos digitales que soportan la comunicación sobre el protocolo IP y que pueden emplear protocolos para aplicaciones en tiempo real como el RTP (*Real Time Protocol*) [RFC1889].

Así, el servicio de VoIP sustituye el tradicional teléfono residencial por un teléfono de gran ancho de banda que utiliza *Internet* para hacer y recibir llamadas. Si se realiza una llamada a un número de teléfono tradicional, la señal es convertida en el otro extremo. Dependiendo del tipo de servicio VoIP, se puede hacer una llamada VoIP desde el ordenador, desde un teléfono especial VoIP o desde un teléfono tradicional con o sin adaptador.

A continuación enumeramos las distintas funciones que caracterizan VoIP y por las que resulta tan ventajoso su uso. En primer lugar, permite realizar más de una llamada telefónica simultáneamente. Además, las llamadas entrantes pueden ser automáticamente dirigidas al teléfono VoIP independientemente de dónde se esté conectado a la red. Muchos operadores de VoIP incluyen servicios gratuitos que en las PSTN (*Public Switched Telephone Network*, Red Pública de Telefonía Conmutada) la mayoría de las compañías de telecomunicación normalmente cargan como una tarifa extra. Ejemplos pueden ser una llamada a tres, llamadas al extranjero, re-llamada automática o identificador de llamada.

VoIP puede ser más segura ya que permite el uso de protocolos como el *Secure Real-time Transport Protocol* (SRTP, Protocolo de Seguridad de Transporte en Tiempo Real) [RFC3711]. Sólo es necesario cifrar y autenticar la trama de datos. Asimismo, VoIP es independiente del lugar, sólo es necesaria una conexión a *Internet* para conseguir una conexión a un proveedor VoIP. Los teléfonos VoIP pueden integrarse

con otros servicios disponibles en *Internet*, incluyendo conferencia de audio o de video, mensajes o intercambio de archivos de datos en paralelo con la conversación. Finalmente, habilita la transmisión en un mayor ancho de banda que puede significar mejorar la calidad de la conversación y la música.

2.2.1 Ventajas de una red VoIP

Vamos a enumerar las ventajas de la VoIP frente a las centralitas privadas de telefonía (PBX, *Private Branch Exchange*). Tradicionalmente, las empresas optan por este tipo de equipos, de ahí que mencionamos también en este apartado las ventajas que VoIP puede ofrecer desde el punto de vista empresarial.

Con respecto a una PBX, VoIP presenta la ventaja del acceso abierto. La documentación en un estándar abierto, así que se puede escribir o modificar código sin problemas. Mientras, en una PBX no hay portabilidad para las aplicaciones que el usuario quiera crear. Por otro lado, si queremos trasladar una cantidad de departamentos a una nueva planta manteniendo la numeración, con VoIP sólo es necesario desplazar los teléfonos, conectarlos a la red y registrarlos con el servidor que dirige las llamadas. En una PBX la gestión es mucho más compleja, pudiendo incluso suponer modificar el cableado interno de la misma. Con VoIP la movilidad es completamente transparente.

Desde el punto de vista de las empresas, está comprobado que VoIP realiza la productividad, ocasiona costes operacionales más bajos y un incremento visible de la gestión/administración, ya que los usuarios necesitan una sola red para transmitir voz y datos. Según Cisco [CISC07] el uso de VoIP para la realización de conferencias reduce en un 30% el gasto de su realización. Las aplicaciones de VoIP debido a que se encuentran en continuo desarrollo, proporcionan ininterrumpidos servicios de calidad, fiabilidad, seguridad y compatibilidad para teléfonos basados en IP, video y convergencia a aplicaciones de escritorio. Tal y como mencionamos antes con VoIP se pueden recibir y enviar llamadas independientemente de la ubicación con sólo conectarse a una conexión de banda ancha (ADSL, Cable, WiFi, etc.).

Otra ventaja importante desde el punto de vista empresarial que aporta la mayoría de aplicaciones, entre ellas Skype [SKYPE08], es que las llamadas a otros usuarios de VoIP son gratuitas y las realizadas a fijos o móviles a nivel mundial se facturan por segundos y sin establecimiento de llamada. En la mayoría de las aplicaciones no existe ninguna cuota de alta ni mensual a diferencia del sistema telefónico normal, con el ahorro que esto conlleva. Señalar, que si una empresa dispone de varias delegaciones tanto a nivel nacional como internacional puede establecer la comunicación sin coste alguno, usando programas de VoIP.

La mayoría de aplicaciones también permiten correo de voz y además de forma gratuita, así como tienen soporte para recibir correctamente mensajes privados desde una conferencia multiusuario. La mayoría de las aplicaciones ofrece soporte para el envío de archivos a usuarios con los que se está conversando, como por ejemplo hace Jabbin [JAB08]. Incluso, algunas aplicaciones poseen una herramienta que les permite establecer comunicaciones con otros usuarios de la misma red, pero de otras plataformas VoIP. Otra característica es que algunas aplicaciones permiten el envío de mensajes de texto ya sea de forma gratuita o mediante una pequeña tarifa y la mayoría de aplicaciones ofrecen recibir llamadas desde cualquier teléfono móvil o línea fija.

Como ventaja podemos incluir también el uso de protocolos de encaminamiento dinámicos, de esta manera una red puede converger utilizando una mejor ruta. Además existen aplicaciones que permiten monitorizar la red indicando el uso que se está haciendo de ella y el destino de las llamadas. Esto es una cualidad añadida que

permite a los administradores de las redes controlar en todo momento el uso de la red de la empresa. Se impide que se utilice para otros usos que no sean los estrictamente necesarios en la empresa.

La mayoría de las aplicaciones creadas para las empresas son versiones mejoradas de las proporcionadas a los usuarios individuales. Además, el software es de fácil instalación y en muchos casos se puede instalar en los teléfonos móviles, como es el caso de Gizmo [GIZMO08], para que funcionen a través de WiFi. Supone un ahorro en la telefonía móvil de la empresa, el terminal sólo necesita captar una red WiFi para establecer una comunicación de VoIP sin necesidad de realizar una llamada telefónica a través de la compañía contratada.

2.2.2 Desventajas de una red VoIP

Aunque VoIP presenta muchas ventajas también hay que tener en cuenta algunas desventajas que se pueden presentar:

1) Necesidad de conexión

Los servicios de VoIP necesitan que la conexión de alta velocidad de *Internet* esté habilitada, si las llamadas se realizan mediante un ordenador entonces debe estar siempre encendido, si se trata de un teléfono especial solo es necesario tener el acceso a *Internet* activo.

2) Páginas blancas

Los proveedores de VoIP podrían no ofrecer un listado de páginas blancas.

3) Fallos de electricidad

Cuando no hay electricidad los proveedores de servicios no tienen una fuente alternativa para mantener el equipo y se apaga, por tanto se establecería un estado de incomunicación.

4) Envío de faxes

El envío de faxes está todavía limitado, la existencia de codificadores de voz no está diseñada para la transmisión de fax. Para remediar esto se podría definir una solución alternativa basada en el desarrollo *Fax-over-IP*, llamado protocolo T.38. Otra posible solución es tratar el sistema de fax como un sistema conmutador de mensajes que no necesita transmitir datos en tiempo real, como un documento de *e-mail* o impresora remota. El sistema final podría completar el buffer de entrada de datos de fax antes de visualizarlos o imprimir la imagen de fax.

5) Llamadas de emergencia

No todos los servicios de VoIP pueden conectar directamente con los servicios de emergencia.

La naturaleza de IP hace difícil localizar a los usuarios de la red geográficamente. No es fácil encaminar las llamadas de emergencia a un centro de llamadas cercano y en algunos sistemas VoIP es imposible. Siguiendo la ventaja de los operadores de telefonía móvil, varias líneas de VoIP son implementadas con una técnica *work-around*, esta técnica implica un arreglo temporal a la espera de una solución auténtica.

Algunos proveedores de VoIP han creado servicios realzados de números de emergencia, aunque el problema surge cuando algunos de estos servicios no pueden seguir eficazmente la localización del usuario debido a que se trata de una red IP. Hasta que se encuentre una solución adecuada a este problema de

VoIP con los números de emergencia, los proveedores recomiendan mantener un teléfono convencional o un teléfono móvil siempre disponible para realiza este tipo de llamadas.

2.2.3 Comparación entre PSTN y VoIP

Tanto en PSTN como en VoIP la comunicación se realiza mediante cuatro funciones básicas. A continuación se muestra la diferencia existente entre las distintas funciones debido al modo en el que se transmitan:

Servicios de Bases de Datos

Las redes PSTN y VoIP la usan para localizar los puntos a los que les tienen que dar la llamada, las PSTN identifican los puntos finales por un número de teléfono y las redes VoIP los identifican por su dirección IP y por su número de puerto, algunas usan DNS (*Domain Name System*, Sistema con Nombres de Dominio) [RFC1035] para direcciones abstractas.

Señalización

En una PSTN los teléfonos se comunican con multiplexación por división en el tiempo (TDM, *Time Division Multiplexing*) [RFC4553] mediante conmutadores (*switches*) o PBX.

En una red VoIP, unos componentes se comunican y establecen sesiones con otros mediante el intercambio de mensajes de datagrama IP, su formato es realizado por cualquiera de los protocolos estandarizados, los más comunes son H.323 [ITUH.323], SIP (*Session Inicial Protocol*, Protocolo de Inicio de Sesiones) [RFC3261], MGCP (*Media Gateway Control Protocol*, Protocolo de Control de Dispositivos) [RFC2705] y H.248 [ITUH.248] también conocido como Megaco [RFC2885].

Conexión de llamada y mecanismos de transporte del audio

En una PSTN los *switches* completan las llamadas conectando canales *Logical Digital Signal-0* (DS0, Señal Digital) a través de la red, cada DS-0 es un canal bidireccional de 64 Kbps que está dedicado exclusivamente a la sesión de comunicación durante la llamada, se usa *Pulse Code Modulation* (PCM, Modulación por Impulsos Codificados) [ITUPCM] para representar las frecuencias analógicas de audio, habilitando a la red a transmitir audio a través de canales DS-0 como valor de pulsos de amplitud de codificación digital.

VoIP usa PCM para codificar el audio. Transmite el audio a través de un canal dedicado DS-0, pero usando recursos de redes compartidos. Estas redes colocan uno o más muestras de PCM, llamadas tramas, en un datagrama IP. El formato de los datagramas será acorde al protocolo de transporte en tiempo real.

Operaciones de los codificadores

La cuarta función básica de una red es el proceso de convertir las ondas analógicas en información digital. Tanto PSTN y VoIP usan codificadores. Este proceso es complejo, hay muchas formas de transformación como nos muestra la tabla establecida por la *Internacional Telecommunications Union* (ITU, Unión Internacional de Telecomunicaciones) (véase Fig.2.1).

Codificadores estandarizados ITU	Descripción	Ancho de banda (Kbps)
G.711	PCM	64
G.721	ADPCM	32, 16, 24,40
G.722	PCM-ADPCM	48, 56, 64,
G.728	LD-CELP	16
G.729	CS-ACELP	8
G.723.1	CELP	6.3, 5.3

Fig.2.1 Tabla con los codificadores estandarizados establecidos por la ITU.

2.3 Consideraciones de diseño de una red VoIP

2.3.1 Muestrear y cuantificar

Cada muestra representa el estado de cuantificación de un instante específico. La conversión de analógico a digital está caracterizada por la velocidad de muestreo y el método de cuantificación. Matemáticamente el proceso de muestrear se define como el resultado de multiplicar un infinito pulso periódico de amplitud 1 por la señal original de tiempo continuo. Se usa el PAM (*Pulse Amplitude Modulation*, Modulación por Amplitud de Pulso) [ITUPAM] para muestrear la señal como técnica más común.

Según el teorema de Nyquist (teorema de muestreo), para procesar una señal continua con componentes frecuenciales comprendidos entre cero y una frecuencia máxima, la velocidad de muestreo debería ser al menos del doble de la frecuencia máxima. Para cuantificar debemos medir la amplitud de cada pulso. Dependiendo de la precisión de la escala, el número que representa la señal PAM puede ser más o menos preciso, pero nunca será exacto.

Existen dos fuentes de pérdida de información cuando preparamos una señal para el proceso digital:

- la pérdida de componentes de alta frecuencia
- la cuantificación del ruido

Las redes de telefonía IP tienen restricciones de calidad de voz debido a que existen diversas circunstancias por las que la red sufre latencias, con respecto la voz a tiempo real, o pérdidas de voz. Para evitarlo lo principal que se le tiene que dar es:

1. eco de control correctamente
2. minimizar la degradación introducida por los paquetes perdidos
3. mantener buena interactividad de velocidad

En los primeros sistemas VoIP, estos tres factores fueron un problema y desafortunadamente trajo una mala percepción de VoIP que todavía persiste hoy:

- **Paquetes perdidos y retraso:** en 1998 *Internet* tenía muy poca capacidad instalada para el tráfico IP, además las redes IP fueron construidas en redes *frame relay*, que introducen largos retrasos. La situación cambió después del “boom” de *Internet*. Los principales portadores del tráfico IP hoy usan las tecnologías más novedosas, como MPLS (*Multiprotocol Label Switching*, Conmutación Multi-Protocolo mediante Etiquetas) [RFC3032], y son capaces de manejar grandes volúmenes de tráfico con un retraso

mínimo. Como el tráfico IP domina a cualquier otra clase de datos se usan menos capas de encapsulación para transportar los paquetes IP. La capa de transporte de *frame relay* ha sido completamente abandonada, incluso la intermediaria capa de transporte ATM está empezando a ser un problema para la velocidad. Por ejemplo, muchos *backbones* llevan ahora los paquetes IP directamente sobre SDH (*Synchronous Digital Hierarchy*, Jerarquía Digital Síncrona). Finalmente, la mayoría de las pasarelas (*gateways*) VoIP y teléfonos IP implementan sofisticados métodos de paquetes perdidos que pueden eficientemente enmascarar dos tramas perdidas consecutivas, haciendo el paquete perdido menos perceptible.

- **Eco de control:** las primeras implementaciones de las *gateway* VoIP, no implementaban satisfactoriamente la cancelación del eco, provocando con ello distorsiones y ruidos. Con la consolidación del mercado de las telecomunicaciones, la mayoría de los *gateway* manufacturados son equipos vendedores DCME (*Digital Circuit Multiplication Equipment*, Equipos Multiplicadores del Circuito Digital) con extensos *know-how* en el procesamiento de la señal y cancelación del eco, o compañías usando la referencia a los algoritmos propuestos por sus vendedores DSP (*Digital Signal Processor*, Procesador Digital de la Señal). Como resultado la calidad de la cancelación del eco en los *gateways* de VoIP y los teléfonos IP es ahora mucho mejor, alcanzando el nivel de atenuación del eco necesario para que el retraso de la voz en las redes no suponga una distracción o inconveniente en la comunicación.

2.3.2 Implementación

Como UDP (*User Datagram Protocol*, Protocolo de transporte basado en Datagramas) [RFC768] no proporciona un mecanismo para asegurar que los paquetes de datos son enviados en orden secuencial ni garantiza una calidad de servicio, las implementaciones de VoIP tienen problemas con la latencia y el *jitter*. Para evitar este problema el nodo receptor debería reestructurar los paquetes IP que lleguen desordenados, retrasados o perdidos, mientras que asegura que la trama de audio mantiene una temporización adecuada. Esta funcionalidad es realizada por un elemento denominado *buffer de jitter*. Otro cambio posible a realizar para evitar los problemas que surgen con UDP es dirigir el tráfico VoIP a través de cortafuegos y traductores de direcciones.

Los retos de VoIP a los que tiene que hacer frente y que analizamos a continuación son:

- Ancho de banda disponible
- Retraso/latencia
- Paquetes perdidos
- *Jitter*
- Eco
- Seguridad
- Fiabilidad
- Traducción del pulso de marcación a DTMF (*Dual-Tone Multi-Frequency*, Tonos de varias Frecuencias).

Ancho de banda disponible

Uno de los requisitos más importantes para aportar calidad de servicio a una aplicación de VoIP es proporcionar suficiente ancho de banda a la red para que pueda soportar el tráfico de voz en tiempo real.

Los factores principales de los que depende el ancho de banda son del codificador usado y el número de muestras de voz que se deseen introducir por paquete. La combinación de estos requisitos puede provocar un uso excesivo del tamaño del ancho de banda, se necesitaría un uso exclusivo del canal sólo para transmitir paquetes de VoIP. Sin embargo, existe una relación que determina que la calidad de los datos transmitidos suele ser directamente proporcional a la cantidad de ancho de banda utilizado. Entre los codificadores utilizados en VoIP encontramos los G.711, G.723.1 y el G.729 (especificados por la ITU-T).

La decisión de la cantidad de ancho de banda asignado es una determinación importante. Se tienen que considerar las prioridades que cada organización determine, el ancho de banda disponible y el coste dispuesto a usar para este servicio. Los principales factores que se tienen que considerar son:

- El impacto de la prioridad de ancho de banda: el tráfico de voz recibe mayor prioridad que cualquier tráfico de datos, debido a la naturaleza de los paquetes de VoIP.
- El intercambio entre compresión y alta calidad: la tecnología de los codificadores puede comprimir paquetes de VoIP y reducir la cantidad de ancho de banda. Sin embargo la compresión disminuye la calidad de voz de la llamada. Como consecuencia se tiene que equilibrar el ancho de banda ahorrado con las consideraciones de calidad requeridas.
- El ancho de banda máximo que se calcula que va a ser usado: los administradores de redes asignan el ancho de banda de la red basándose en un número máximo de llamadas durante las horas de pico. Además también se tiene que considerar el ancho de banda necesario para señalización

El ancho de banda necesario para señalización depende principalmente de la velocidad a la cual las llamadas son generadas y el protocolo de señalización usado. Si un gran número de llamadas son iniciadas en un corto periodo de tiempo, el pico de ancho de banda necesario para señalización podría ser demasiado alto. En general, se puede determinar que la máxima cantidad de ancho de banda necesario por un protocolo de señalización IP es 3% del total tráfico [JUNI07].

Retraso/Latencia

La latencia es un parámetro que afecta a las transmisiones realizadas mediante VoIP. Existen dos tipos de retardo:

- Los retrasos fijos no pueden ser controlados debido a que las características de *Internet* varían. Se producen en los procesos de transmisión del paquete por los distintos dispositivos de la red, como pueden ser: la codificación, la creación de paquetes, la serialización y buffer de jitter.
- Los retrasos variables son los producidos debido a las colas de salida y a los retrasos de la red.

El tráfico de voz es un tráfico en tiempo real, un gran retraso en la entrega de los paquetes de voz provoca que la conversación no sea entendible. Se puede determinar que un retraso aceptable tiene que ser inferior a 150 ms. Este retardo máximo total se distingue en tres tipos de retraso que son inherentes en las redes de telefonía:

- Retraso de propagación es causado por la velocidad de la luz viajando por un medio de fibra óptica o por un medio de cobre.

- Retraso del *buffer* de *jitter* transforma los retrasos variables a retrasos fijos ya que deja colgada la primera muestra que recibe durante un periodo de tiempo antes de enviarla. La contribución del *buffer* del *jitter* al retraso es que a parte del retraso propio de *jitter* que la trama posee se le está añadiendo otro retraso debido a este *buffer*.
- Retraso de serialización es causado por los aparatos que manejan la información de voz, es la cantidad de tiempo que tarda en ponerse un *bit* o *byte* en un interfaz de red. Este retraso es directamente influido por la velocidad del reloj del interfaz y por el tamaño del paquete. Se puede modificar el retraso de serialización de un paquete de voz realizando lo siguiente:
 - Incrementando la velocidad del enlace: es una solución cara
 - Disminuyendo el tamaño de los paquetes: puede no ser posible para todos los tipos de codificadores, su mayor desventaja es que aumenta el ancho de banda requerido para cabeceras.
- Retraso de procesamiento incluye muchas diferentes causas de retraso, (paquetización, codificación y conmutación de paquetes) y es causado por aparatos que transmiten las tramas a través de la red.
 - Retraso de cola: en una red basada en paquetes se retrasa por el tiempo necesario para mover el paquete actual a la cola de salida. El retraso de cola ocurre cuando son enviados más paquetes de los que el interfaz puede manejar en un intervalo determinado. Como la voz debería tener absoluta prioridad en el *router* (encaminador), una trama de voz debe esperar solo durante una trama de datos que ya esté sirviéndose o durante tramas de voz que estén anteriormente a ella. Este tipo de retraso es variable y depende de la velocidad de enlace y del estado de las colas.
 - Retraso de reenvío del paquete: tiempo que tarda un aparato de la red (*router*, *switch*, *firewall*) en almacenar un paquete y reenviarlo.
 - Retraso de codificación: también llamado retraso de procesamiento, es el tiempo que se tarda en codificar una secuencia analógica a unas muestras moduladas según un código de pulso. Como cada codificador trabaja distinto, este retraso varía con el codificador de voz que se esté usando y la velocidad del procesador (véase Fig.2.2).

Codificadores estandarizados ITU	Velocidad de bit (kbps)	Tamaño de la trama (ms)	Retraso en la conversión (ms)
G.711 PCM	64	0.125	5
G.729 CS-ACELP	8	10	15
G.729a CS-ACELP	8	10	15

Fig.2.2 Retardo introducido por los codificadores de voz

Compensar los paquetes perdidos

La pérdida de paquetes puede ocurrir por muchas razones, en algunos casos es inevitable, pero la principal causa es debido a la congestión de la red. Durante la congestión los *routers* y *switches* pueden sobrepasar sus colas y descartar paquetes. Otra causa que provoca la pérdida de paquetes es un exceso de latencia, un grupo de

paquetes llega tarde y deben ser descartados en favor de los nuevos. Los paquetes perdidos de las aplicaciones que no son de tiempo real no son críticos. Estas aplicaciones suelen usar TCP (*Transmisión Control Protocol*, Protocolo de Control de Transmisión) [RFC 793] que permite retransmisión, no supondría un gran problema. Sin embargo, las aplicaciones basadas en tiempo real como UDP son menos tolerantes a la pérdida de paquetes porque no tienen posibilidad de retransmisión.

La consecuencia que provoca la pérdida de paquetes es que se producen vacíos en la conversación. La ventaja que tienen los paquetes de VoIP es que su longitud es corta, la pérdida de esta pequeña cantidad de diálogo no empeora la conversación de los interlocutores. Pero probabilísticamente la pérdida de un paquete significa la pérdida de varios paquetes, este hecho sí degrada severamente la calidad del servicio de una red VoIP.

Según un estudio de la Asociación de la Industria de las Telecomunicaciones (TIA, *Telecommunications Industry Association*) muestra que incluso un pequeño porcentaje de paquetes perdidos podría disminuir la QoS (*Quality of Service*, Calidad del Servicio) [RFC2212] de las redes de VoIP según el nivel que los usuarios tienen como esperado de las líneas telefónicas convencionales. Se determina que un porcentaje tolerable de pérdida de paquetes se centra entre 1% y 3% y la calidad empieza a ser intolerable cuando más del 3% de los paquetes de voz se han perdido.

Jitter

Se define como la variación del tiempo de llegada entre paquetes. Mientras el que envía está esperando transmitir fielmente los paquetes de voz a un intervalo regular, debido a los problemas que se pueden producir en la red, como son la congestión, la longitud de las colas demasiado largas o los errores, estos paquetes de voz pueden retrasarse a través de la red y no llegar con el mismo intervalo a la estación receptora. Este hecho provoca una discontinuidad en la trama de la voz de tiempo real. El efecto del jitter afecta al retraso y a la pérdida de paquetes, como se comentó anteriormente, puede ser mitigado si se van guardando los paquetes de voz en un *buffer*.

Las aplicaciones de voz son muy susceptibles a estas variaciones ya que es un parámetro que degrada cuantiosamente la calidad de la voz y modifica el diálogo. Lo que influye directamente en la QoS de la VoIP. El *jitter* se produce normalmente en redes que tienen un ancho de banda de baja capacidad. Puede causar que los paquetes a su llegada al receptor sean procesados fuera de secuencia.

El *jitter* puede incluso afectar en mayor medida a la QoS que el propio retraso de los paquetes. Si el *jitter* es alto, los paquetes llegan a su destino a ráfagas. Cuando los *routers* reciben una trama de VoIP, compensan el *jitter* que encuentran. Tienen que ser capaces de almacenar en un *buffer* estos paquetes y reenviarlos con el mismo espaciado entre cada uno de ellos, para reducir el *jitter*. Si la magnitud de *jitter* es tan grande que los paquetes son recibidos fuera del rango del *buffer*, entonces son descartados provocando la pérdida de paquetes.

Podemos controlar el problema del *jitter* mediante el uso de *routers*, *firewalls* (cortafuegos) y otros elementos de la red que puedan soportar QoS.

Eco

El eco más importante es el producido cuando el interlocutor habla, la percepción para el hablante es que escucha su propia voz pero retrasada. Esto puede ser causado por el eco eléctrico o por el acústico.

Si el eco se refleja dos veces, puede afectar a la persona que está escuchando en el otro extremo de la comunicación. En un inusual caso se produciría que la persona escucha la voz del que habla varias veces.

Existen dos tipos de eco el eléctrico y el acústico:

Eco eléctrico

Se produce cuando se realiza un acoplamiento acústico de la señal transmitida y recibida en el receptor final. Si excede 25 ms puede provocar distracciones y romper la conversación. El hablante al final de la otra línea se escuchará a si mismo después de un periodo de tiempo.

Eco acústico

Normalmente el eco acústico es una señal parásita entre 10 y 15 dB que se une a la señal acústica de la persona que esté hablando. Este eco es inapreciable si el retraso que la señal sufre no supera los 50 ms. Si aumentamos este tiempo la persona que está hablando percibirá su propia voz.

Una manera para eliminar el eco acústico es usar cascos. Sin embargo, con mecanismos adecuados que cancelen el eco es posible reducir la potencia de la señal de eco.

Limitar el eco

Existen dos tipos de dispositivos que normalmente son usados para limitar el eco: los canceladores de eco y los supresores de eco.

- Supresores de eco: consiste en introducir grandes pérdidas en la transmisión mientras el emisor está hablando. El sonido de fondo que se percibe sobre la voz del hablante, de repente desaparece cuando él para de hablar o cuando el que estaba escuchando comienza a hablar.
- Canceladores de eco: son modelos funcionales. Este eco es modelado como la suma de las similares señales de entrada, pero retrasadas y con una baja amplitud. Técnicamente los canceladores de eco son filtros FIR (*Finite Impulse Response*, Finita respuesta al Impulso), encargados de seleccionar únicamente de la señal de salida la porción que les interesa para eliminar los armónicos que producen al eco.

Seguridad

Las redes de VoIP son vulnerables a muchos riesgos de seguridad: ataques de denegación de servicio (DoS, *Denial of Service*), ladrones de servicio, *tampering* y fraude entre otros. Para proporcionar las soluciones VoIP, el aparato de seguridad debe ser capaz de soportar los protocolos de VoIP como son SIP, MGCP y H.323 y asociar el estado de la capa de señalización con el flujo de paquetes en la capa entre enlaces.

Muchas aplicaciones de VoIP no soportan todavía cifrado, aunque tener un teléfono seguro es mucho más fácil de implementar en VoIP que en las tradicionales líneas telefónicas. Algunos vendedores solo usan compresión para hacer más difícil las escuchas no deseadas. Sin embargo, la seguridad real requiere cifrado y autenticación que no estén ampliamente disponibles al nivel de consumidor. Como ejemplo, la existencia del estándar de seguridad SRTP y del nuevo protocolo ZRTP [ZRTP] está disponible en un ATA (*Analog Telephone Adapters*, Adaptador de Teléfono Analógico) y en varios programas de *software* que permiten hacer llamadas a través de *Internet* mediante el ordenador.

Los encargados de las redes podrían emplear el protocolo de autenticación construido en IEEE 802.1x llamado EAPOL (*Extensible Authentication Protocol over*

LANs, Protocolo de Autenticación Extensible sobre LANs). 802.1x basado en EAPOL ofrece más beneficios que la mayoría del resto de sistemas de autenticación. Podríamos enumerar por ejemplo el acceso de control a la red (NAC, *Network Access Control*) con una vía de entrega segura y fuerte; una estructura flexible para la autenticación, la seguridad de la red y la privacidad de los datos; y la creación de un poderoso entorno defensivo de la red, mediante los robustos controladores que permiten entrar a la red de la empresa a sólo los usuarios que tengan unas características determinadas. También completa la autenticación de la red antes de que la dirección IP sea asignada asegurándose que los virus hayan desaparecido antes de que se propaguen por la organización.

Los administradores de la red deberían establecer políticas de cortafuegos para proteger la comunicación entre servidores y los clientes de VoIP. Estas políticas deberían restringir la comunicación VoIP basada en un cliente autorizado o fuente de tráfico o destino con una determinada dirección IP o interfaz. Es importante porque el 80% del origen de los ataques se produce dentro de la red [JUN107]. Los cortafuegos pueden segmentar las redes VoIP, separando el tráfico de voz de otro tráfico para asegurar que se están aplicando las prioridades y políticas. Los cortafuegos también pueden ayudar a mitigar los ataques DoS.

Además se pueden desarrollar medidas preventivas para evitar ataques como la manipulación de los mensajes de *Dynamic Host Configuration Protocol* (DHCP, Protocolo de Configuración Dinámica en el Host) o la inundación de las tablas de encaminamiento.

Fiabilidad

Es esencial planear los posibles fallos de la red, aunque se produzcan pocas veces. Los administradores de las redes necesitan realizar estrategias de fallo para cuando los aparatos de la red fallen o los enlaces se rompan. Para prever estas situaciones se deben establecer enlaces redundantes entre los aparatos de la red y equipos redundantes.

Las redes IP usan protocolos de encaminamiento para intercambiar la información de la ruta. Estos protocolos detectan y redirigen los paquetes a un camino alternativo si se produjese un fallo. El tiempo necesario para detectar y recalculan un camino alternativo puede depender de la interconexión para esos enlaces.

Otro problema que afecta a la fiabilidad de la aplicación es que el *hardware* de VoIP usa *módems* de ancho de banda y otros equipos de energía que dependen de la electricidad del hogar para su funcionamiento. Debe usarse un generador supletorio para asegurar la fiabilidad durante las caídas de energía.

Traducción del Pulso de Marcación a DTMF

Muchos proveedores de VoIP no traducen el pulso de marcado de los viejos teléfonos a DTMF. El usuario de VoIP podría usar un *VoIP Pulse to Tone Converter* (Convertidor de Tonos a Pulsos) si fuera necesario realizar esta traducción.

2.4 Mecanismos que garantizan la Calidad de Servicio

Internet es una red “*best effort*” esto significa que los nodos de la red intentan enviar los paquetes a su destino, pero no garantizan que los paquetes sean enviados, así como no garantizan el ancho de banda utilizado, el retraso o el *jitter* que puedan sufrir los paquetes. Para garantizar que estos parámetros queden dentro de unos

límites establecidos, se han de usar junto con IP alguna de las arquitecturas de Calidad de Servicio (por ejemplo Servicios Integrados o Servicios Diferenciados).

La calidad del servicio, fue ignorada en los primeros diseños de las redes IP. IP como otras tecnologías de redes de paquetes, fue construida y optimizada para transportar archivos de datos, no voz o vídeo. Por lo tanto la única calidad del servicio necesaria para ello era que los datos no fueran corruptos o existieran pérdidas. Hoy en día, el continuo desarrollo y mejora de la tecnología de las redes hacen posible el transporte de datos en tiempo real, como son la voz o el vídeo.

Sin embargo, no sólo es importante transportar los datos, también hay que proporcionarles una calidad de servicio adecuada. Los principales parámetros que proporcionan este requisito en una red de paquetes son:

- la capacidad disponible, también llamado ancho de banda
- el retraso en la transmisión de un punto a otro y su variación (*jitter*)
- los paquetes perdidos y desordenados.

Básicamente la calidad de servicio en las redes de voz sobre IP se intenta obtener mediante dos mecanismos: *Integrated Services* (IntServ, Servicios Integrados) [RFC1633] y *Differentiated Services* (DiffServ, Servicios Diferenciados) [RFC2740].

IntServ es normalmente usado para asegurar que un específico flujo de tráfico va a recibir unos determinados niveles de QoS a través de la red antes de enviar este tráfico. Esto es logrado usando el protocolo RSVP (*Resource ReserVation Protocol*, Protocolo de Reserva de Recursos) [RFC2205].

DiffServ es normalmente usado en grandes redes para clasificar los niveles de QoS de un determinado fragmento. Esto se consigue mediante el campo ToS (*Type of Service*, Tipo de Servicio) [RFC791] o DSCP (*Differentiated Service Code Point*, Código de Diferenciación de Servicios) [RFC2474].

Principalmente existen dos factores que exigen estos requisitos se cumplan y se satisfaga la calidad de servicio de las aplicaciones, y son:

- las técnicas necesarias para la comunicación de voz en tiempo real
- la alta expectación de los usuarios: acostumbrados a la calidad proporcionada por las PSTN y por las redes privadas basadas en PBX. Éstas utilizan conexiones orientadas donde cada usuario tiene un ancho de banda dedicado para la duración de cada llamada. Dando como resultado unos valores muy bajos en latencia y en *jitter* y una mínima interrupción debida al ruido de la conexión.

A continuación enumeraremos los elementos que influyen en la calidad de servicio de una aplicación que transmite paquetes de VoIP.

2.4.1 Codificadores y decodificadores

Los codificadores son elementos que influyen en la calidad de la voz, debido a que están directamente relacionados con el retardo. La forma en la que actúan es que los codificadores en los terminales de VoIP comprimen la señal de voz e introducen tres tipos de retraso:

- procesamiento: tiempo necesario para codificar el código en una sola trama de voz
- mirar el retraso futuro: el tiempo necesario para un codificador para examinar la parte de la trama siguiente mientras codifica la trama actual

- retraso de la trama: tiempo necesario para que el sistema que envía transmita la trama

La siguiente figura (Fig.2.3) muestra los métodos de compresión de señal y sus características. Podemos apreciar que una de sus columnas es la de puntuación MOS (*Mean Opinion Score*, Puntuación de la Opinión Principal) [ITUP.800], lo que mide es la puntuación de las opiniones sobre ese codificador (1 es malo hasta 5 excelente).

Codificadores estandarizados ITU	Velocidad de bit (kbps)	Tamaño de la trama (ms)	Puntuación MOS
G.711 PCM	64	0.125	4.1
G.729 CS-ACELP	8	10	3.92
G.729 x 2	8	10	3.27
G.729 x 3	8	10	2.68
G.729a CS-ACELP	8	10	3.7

Fig.2.3 Tabla que muestra los métodos de compresión y su puntuación (MOS Score)

G.711

El G.711 es un estándar de la ITU-T para la compresión de audio. Este estándar es usado principalmente en telefonía y fue liberado para su uso en el año 1972.

G.711 es un estándar para representar señales de audio a frecuencias de la voz humana, mediante muestras comprimidas de una señal de audio digital con una tasa de muestreo de 8000 muestras por segundo. El codificador G.711 proporciona un flujo de datos de 64 kbit/s.

G.722

Para Voz IP hay grandes requerimientos para ofrecer mejor velocidad y calidad de audio para los sistemas de videoconferencias y audio conferencias. G.722 fue propuesto por *France Telecom* y NTT (*Nippon Telegraph and Telephone*, Telefonía y Telegrafía Japonesa) y adoptado por ITU en 1988. La idea fundamental es repartir el ancho de banda que tiene que ser transmitido en 2 sub-bandas:

- una sub-banda baja de 0 Hz a 4000 Hz
- una sub-banda alta de 4000 Hz a 8000 Hz

Después de un nuevo proceso de muestreo para reducir la frecuencia de muestreo de 16 kHz a 8 kHz, es posible volver a muestrear porque la frecuencia de sub-banda al ser filtrada ha eliminado el efecto de *aliasing*.

G.722 codifica las señales en tramas de bit de 64 kbit/s. Se trata del PCM básico. En la sub-banda baja, 6 bits son usados para adaptar la cuantificación con una incrustación característica: el cuantificador central usa 4 bits y la versión realzada usa 6 bits. Esto permite al sistema disponer de algunos bits de los propósitos de la señalización (trama con H.221) para transmitir algunos datos auxiliares. En el decodificador debería ser señalado el modo de operación (64, 56 o 48 kbit/s), aunque algunas realizaciones no señalizan el modo de operación y continuamente utilizan

lentos los 6 bits. En la sub-banda alta, 2 bits del cuantificador adaptado son usados para producir una velocidad de 16kbit/s.

El codificador de voz de ancho de banda de la ITU-T G.722 es comúnmente usado en sistemas de teleconferencias cumpliendo las recomendaciones H.320. La calidad de la codificación es calificada como buena para la voz y música a 64 kbit/s y 56 kbit/s.

G.722 comparte con otras formas de onda tipos codificados ADPCM una gran inmunidad a los errores de bit y es más robusto que una trama directa PCM. La gran ventaja del G.722 es que proporciona menor retraso. Los filtros de análisis y síntesis de QMF añaden un retraso de 3 ms. El resultado total del retraso permanece excelente y asegura una buena interactividad para los sistemas de teleconferencias.

Uno de los inconvenientes de G.722 es que no puede ser usado por debajo de los 48 kbit/s. Sin embargo existe una versión más reciente, la G.722.1 capaz de codificar una señal con una velocidad de bit de 24 kbit/s o 32 kbit/s.

G.723.1

G.723.1 es un codificador de audio que comprime la voz en 30 ms cada trama. Existen dos velocidades de bit a la cuales puede operar:

- 6.3 kbit/s (usando tramas de 24 bytes) utiliza el algoritmo MPC-MLQ (*Multipulse LPC with Maximum Likelihood Quantization*, Multipulso LPC con la Máxima Probabilidad de Cuantificación)
- 5.3 kbit/s (usando tramas de 20 bytes) usando el algoritmo ACELP (*Algebraic Code Excited Linear Prediction*, Predicción Linear del Código Algebraico Excitado)

La frecuencia de muestreo que utiliza es 8 kHz. El retraso que introduce este codificador es de 37.5 ms por trama si tenemos en cuenta los 7.5 ms de retraso que genera el mirar la cabecera. Se trata de un algoritmo bastante complejo. La patente de este algoritmo expira en 2014, hasta entonces se necesita una licencia para poder usarlo.

G.728

G.728 es un estándar ITU-T para la compresión de voz, opera a 16 kbit/s y la tecnología que usa es LD-CELP (*Low Delay Code Excited Linear Prediction*, Predicción Lineal del Bajo Retraso del Código). El retraso que tiene este estándar es de 5 muestras lo que equivale a 0.625 ms.

G.729

G.729 es un estándar ITU-T. Se trata de un algoritmo de compresión de datos de audio para voz, comprime audio de voz en trozos de 10 ms. La música o los tonos, como por ejemplo los de fax, no pueden ser transportados fiablemente en este algoritmo.

Generalmente, se usa en aplicaciones de VoIP porque necesita poco ancho de banda. Este estándar opera a una tasa de bits de 8 kbit/s, aunque también podemos encontrar tasas de 6.4 y 11.8 kbit/s.

G.729.1 es la versión extendida de G.729 que se ha realizado para dar soporte a las conversaciones de banda ancha y codificación de audio, el rango de frecuencia acústica se extiende a 50 Hz-7 kHz.

G.729a es otra de las extensiones de G.729, se diferencian en que la primera necesita menos cómputo lo que implica que la calidad de la conversación está empeorada.

2.4.2 Clasificación de los paquetes

Utilizando la clasificación de los paquetes podríamos asegurar que se cumple la calidad de servicio esperada para la voz, para ello tendríamos que saber clasificar los distintos tipos de paquetes y según este criterio aplicar unas ciertas prioridades a cada uno con respecto a los anteriores. Nuestro objetivo es maximizar la calidad de servicio de los paquetes de voz, para conseguir este efecto se le debería dotar a estos paquetes de mayor prioridad en cola con respecto a todos los demás. De esta manera estaríamos proporcionando un retardo mínimo para los paquetes de voz.

Un posible criterio a aplicar para la clasificación de los paquetes es identificar y agrupar los procesos según su dirección IP fuente y/o destino, estos datos son proporcionados por la cabecera IP de la capa 4. O también podríamos emplear el número de puerto fuente y/o destino de UDP, que encontraríamos en la cabecera de la capa 3.

Otro posible criterio sería usar mecanismo de clasificación dinámica como puede ser el RSVP. Este mecanismo usa la señalización de paquetes H.245 [ITUH.245] para determinar que puerto UDP será usado para la conversación de voz. Este mecanismo identifica el tráfico de VoIP y lo sitúa en colas reservadas.

No obstante, puede que los inconvenientes de la clasificación de los paquetes sean superiores a las ventajas. Su principal desventaja es que requiere un gran uso de la CPU para escanear cada cabecera de cada paquete según el tipo de clasificación ejecutada.

Un método más eficiente sería realizar la clasificación en la propia red. Este método se realiza mediante el campo ToS es un *byte* que se encuentra en la cabecera IP. Los 3 *bits* más significativos del ToS son llamados *bits* de procedencia IP. Pero si se toman los 6 *bits* más significativos del ToS, llamados DSCP, pueden ser usados para diferenciar tipos de servicios (DS, *Differentiated Services*).

2.4.3 Mecanismos de Cola

Para proporcionar calidad de servicio a los paquetes de voz tenemos que garantizar ancho de banda y servicio prioritario para las colas de salida a través de mecanismos inteligentes. A continuación describimos algunos de los mecanismos más importantes.

FIFO

Este mecanismo de colas llamado *First In First Out* (FIFO, Primero en Entrar es el Primero en Salir) tiene su gran ventaja en su facilidad tanto de operación como de configuración. Consiste en un método que envía a la cola de salida los paquetes en el mismo orden en el que le han llegado. Resulta fácil comprender que su gran inconveniente es que no realiza un servicio de prioridades ni garantiza un ancho de banda. Podríamos concluir que este método es poco útil si queremos proporcionar una alta calidad de servicio para VoIP.

WFQ

Weighted Fair Queueing (WFQ, Colas según Peso), el peso se determina según la dirección fuente y destino, el tipo de protocolo, socket o número de puerto y según los valores de ToS. Este algoritmo consiste en clasificar los paquetes en distintas colas

según su peso, así determinamos el número de paquetes que son servidos a la misma vez. Las tramas que tienen poca cantidad de datos tienen preferencia en la transmisión mientras que las tramas cuya cantidad de datos es muy grande comparten la capacidad con las anteriores. De esta manera cada uno obtiene un ancho de banda proporcional.

Sus ventajas son que tiene una configuración muy simple, reduce el *jitter* y proporciona un uso proporcional del canal entre todas las aplicaciones. Sin embargo, el inconveniente es que no proporciona servicio prioritario ni garantiza un ancho de banda fijo.

CQ

El método *Custom Queueing* (CQ, Poner en Cola según la Clase). El tráfico se clasifica en múltiples colas, hasta 16 se pueden asignar, cada cola se establece según unos límites. Estos límites son calculados haciendo una media del tamaño de paquete, la unidad máxima de transmisión (MTU, *Maximum Transmission Unit*) y el porcentaje de ancho de banda que sea disponible. Cada cola es servida secuencialmente, se transmite un porcentaje de tráfico que cada cola que tiene almacenado antes de pasar a la siguiente.

Su beneficio radica en que permite disponer de un determinado ancho de banda para cada cola. El inconveniente es que no ofrece prioridad de servicio.

PQ

En el método *Priority Queueing* (PQ, Cola según Prioridad) la funcionalidad es clasificar el tráfico según cuatro colas de distintas prioridades: alta, media, normal y baja. Las colas son servidas de mayor a menor prioridad.

Podemos apreciar que la ventaja es que aporta prioridad de servicio y asegura que al tráfico crítico se le va a ofrecer todo el ancho de banda que necesite.

También nos podemos fijar que el inconveniente se daría en situaciones en las que exista un gran número de paquetes de prioridades altas. En ese momento, el tráfico servido sólo sería el que exista en las colas de mayor prioridad imposibilitando la transmisión de los paquetes que estén en colas de menores prioridades.

Este mecanismo sería útil en las situaciones en las que los paquetes de prioridad alta consuman menos cantidad de ancho de banda.

CB-WFQ

El método *Class-Based WFQ* (CBWFQ, Clases Basadas en WFQ) es usado para clasificar el tráfico en determinadas colas con un reservado ancho de banda cada una según el peso que tengan. Este mecanismo puede manejar hasta 64 clases diferentes, cada una asociada a una determinada cola, y controlar las necesidades de ancho de banda que tengan cada una.

Los beneficios son, además de los que ya tenía WFQ, que garantiza un determinado ancho de banda para cada cola, sin embargo sigue sin proporcionar prioridad en su servicio.

PQ-WFQ

El método *Priority Queue WFQ* (PQ-WFQ, Prioridad de Cola WFQ) es usado para proporcionar prioridad de servicio a los paquetes UDP cuyo puerto destino está especificado en un rango. Su ventaja es que proporciona prioridad a los paquetes RTP, lo que nos resulta muy beneficioso para proporcionar una gran calidad de servicio para los paquetes de VoIP. Sus desventajas son que no garantiza disponer de ancho de banda para otros servicios que son tratados con WFQ y que el tráfico RTCP no lo considera prioritario.

LLQ

Uno de los mecanismos que satisface los requerimientos necesarios para VoIP es *Low Latency Queueing* (LLQ, Baja Latencia en Cola). Este método proporciona prioridad a ciertas clases y garantiza un mínimo ancho de banda para otras clases. Tiene la ventaja que en periodos de congestión en la red configura la velocidad para que el tráfico prioritario no monopolice todo el ancho de banda disponible.

LLQ también se puede configurar para determinar el tamaño de las colas, de esta manera el *router* podría descartar paquetes si hay demasiados esperando en una determinada cola. Existe una clase por defecto (*class default*) que es usada para enviar el tráfico que no está clasificado en cualquiera de las otras colas.

2.4.4 Evitar la congestión.

Como hemos visto anteriormente WFQ, PQ y CQ son mecanismos que en caso de congestión dan prioridad al tráfico importante. Sin embargo más importante que dar prioridad es evitar la congestión. De este modo se impediría descartar paquetes debido a que no se puedan transmitir o almacenar en los buffers de salida. Evitaríamos que el tráfico servido fuera lento, ya que agilizaríamos la transmisión a la velocidad adecuada. Algunos de los métodos que evitan la congestión son:

- Desbordamiento (*Tail Drop*)
- *Random Early Detection* (RED, Detección Anticipada Aleatoria) y *Weighted Random Early Detection* (WRED, Detección Anticipada Aleatoria según la Clase): Se trata de un mecanismo para evitar la congestión que es útil especialmente para redes de gran velocidad.

WRED combina las capacidades del algoritmo RED con *IP Precedence* (Prioridad IP). Esta combinación proporciona que el tráfico preferente sea manejado como paquetes de gran prioridad. Puede seleccionar descartar tráfico cuando los interfaces comienzan a estar congestionados y proporciona la realización de diferentes características para diferentes clases de servicio.

2.4.5 Fragmentación e Inserción

Las transmisiones de VoIP son extremadamente sensibles al retraso, por lo tanto los paquetes de VoIP deben ser insertados entre los fragmentos de los paquetes de datos. De este modo se evita la espera de un paquete de voz hasta que no haya concluido la transmisión de uno de datos.

Hay que tener en cuenta que los paquetes de VoIP tienen que ser enviados cada 20 ms y que las recomendaciones de la ITU-T sugieren no exceder en más de 150 ms el retraso punto a punto para poder mantener una “buena” calidad de voz. Para cumplir estos objetivos tenemos que fragmentar los paquetes de datos para que como máximo tarden 10 ms. E insertar entre estos fragmentos de paquetes los de VoIP.

2.4.6 Desordenación de los paquetes

Debido a que los paquetes de voz son llevados por la red mediante el protocolo de transporte UDP, puede provocar que los paquetes que llegan al otro extremo de la comunicación lleguen desordenados.

Una manera de solucionar este problema sería utilizando el protocolo RTP. Se encarga de proporcionar a los puntos finales de la red funciones que desarrollan servicios para evitar el retraso de los datos en tiempo real como son la voz y el video.

RTP es el protocolo usado para transportar la voz, está basado en UDP, de este modo los paquetes fuera de orden no son ensamblados. Sin embargo RTP permite a las aplicaciones hacer reordenación usando los campos número de secuencia y *timestamp*. Este protocolo será descrito más adelante.

2.5 Protocolos de transporte

Debido a la naturaleza sensible del tráfico de voz, UDP/IP es la elección lógica para transportar la voz, así que para el tráfico en tiempo real o para que fuera sensible al retraso, el *Internet Engineering Task Force* (IETF, Grupo de Trabajo en Ingeniería de Internet) adoptó el RTP. RTP fue diseñado por el IETF's *Audio-Video Transport Working Group* (Grupo de Trabajo de Transporte de Audio y Video de IETF) para soportar conferencias de audio y video con múltiples participantes geográficamente dispersos. De esta manera, VoIP es transportado con una cabecera de paquetes RTP/UDP/IP.

RTP es un protocolo estándar de Internet que especifica una forma para programar y direccionar las transmisiones de tiempo real de datos multimedia sobre servicios de red *unicast* o *multicast* (envío simultáneo de un paquete a varios destinatarios que pertenecen al mismo grupo) teniendo en cuenta el retraso que se produce a través de las redes basadas en paquetes. RTP no garantiza la entrega en tiempo real. Sin embargo, la información en la cabecera RTP dice al receptor como reconstruir los datos y describe como han sido empaquetadas las tramas de *bit* codificadas.

RTP combina su transporte de datos con un protocolo de control RTCP (*Real-time Transport Control Protocol*, Protocolo de Control de RTP) [RFC1889], que hace posible monitorizar los datos entregados a una gran red *multicast*. La monitorización permite al receptor detectar si hay algún paquete perdido y compensar los retrasos de *jitter*. Ambos protocolos trabajan independientemente de los de la capa de transporte y la capa de red.

La parte de datos de RTP es un protocolo que se utiliza en la capa de aplicación. Proporciona soporte para aplicaciones con propiedades en tiempo real (audio y video), incluyendo la sincronización de la reconstrucción, detección de la pérdida y el contenido de la identificación. Los componentes de RTP incluyen:

- número de secuencia: es usado para detectar paquetes perdidos
- identificación *payload*: describe la codificación entre enlaces así que puede ser cambiada si tienes que adaptarla a una variación en el ancho de banda
- indicación de trama: marca el principio y final de cada trama
- identificación de fuente: identifica el origen de la trama
- sincronización: usa *timestamps* para detectar los diferentes retrasos de *jitter* dentro de una sola trama y compensarla por ello.

RTCP proporciona soporte para las conferencias de grupos en tiempo real de cualquier tamaño dentro de Internet. Este soporte incluye una fuente de identificación y soporte para *gateways*, como son los puentes de audio y video tanto como traductores de *multicast* a *unicast*. Los componentes de RTCP incluyen:

- reacción de la calidad de servicio: incluye el número de paquetes perdidos, el tiempo de ida y vuelta y *jitter*, así que las fuentes pueden ajustar su ritmo de datos acordándolo.
- Control de la sesión: usa el paquete BYE de RTCP para permitir a los participantes indicar que están abandonando una sesión
- Identificación: incluye el nombre de un participante, dirección *e-mail*, y número de teléfono para la información de otros participantes

- Sincronización intermedia: habilita la sincronización de los paquetes transmitidos de audio y video separadamente

El uso de RTP es importante para el tráfico de tiempo real pero existen algunos inconvenientes. Las cabeceras IP/RTP/UDP son 20, 8, y 12 bytes respectivamente, esto añade 40 bytes de cabecera. RTP comprimido (CRTP, *Compressed Real-time Transport Protocol*), [RFC2509], fue desarrollado para disminuir el tamaño de las cabeceras IP, UDP y RTP. Fue diseñado para trabajar con enlaces fiables y rápidos punto a punto, pero tiene el inconveniente de que no funciona bien para las aplicaciones VoIP. Existe otra adaptación, *Enhanced CRTP* (ECRTP, Versión Realzada de CRTP) [RF 3545], que fue definida para superar este problema.

Reliable user Data Protocol (RUDP, Protocolo de Datos Fiable) habilita la fiabilidad en una conexión UDP sin la necesidad de un protocolo basado en la conexión como es TCP. El método consiste en enviar varias veces el mismo paquete y habilitar a la estación receptora a descartar los paquetes innecesarios o redundantes. Este mecanismo hace más probable que los paquetes realicen el viaje del emisor al receptor. Los usuarios que tienen casi ilimitado ancho de banda consideran que este es un mecanismo que merece la pena para habilitar la fiabilidad y la calidad de voz.

2.6 Protocolos de señalización

Existen numerosos protocolos de señalización, sin embargo a continuación sólo enumeraremos los más importantes.

H.323

H.323 es una recomendación de ITU para transportar el tráfico multimedia por la red de paquetes. Es un protocolo bastante complejo y necesita de otros protocolos para realizar su función. Es elegido por ITU para establecer conexiones VoIP y define varias funciones de señalización, como son los formatos entre los enlaces de los *routers* relacionados con la paquetización de los servicios de audio y video. Contiene tres componentes de solución principales:

- Servidores de procesamiento de llamada: guarda y aplica información de la topología de la red y de los puntos finales para encaminar las llamadas a los *gateways* de VoIP y a los equipos del usuario final.
- Enlaces *gateways*, sirve tanto para los puntos finales e interfaces de H.323 como con redes que no sean H.323, como PSTN.
- *Gatekeepers*, funciona como una unidad central para controlar la admisión de llamadas, ancho de banda y señalización de las llamadas. Aunque no son necesarios, su uso en las redes incrementa la estabilidad para controlar la separación en llamadas y para dirigir funciones de los *gateways*.

A continuación mostramos la relación que existe entre los componentes de H.323 (véase Fig.2.4).

Las siguientes funciones y capacidades están dentro de un terminal H.323:

- Unidad del Sistema de Control: proporciona control de llamadas H.225 [ITU-T H.225] y H.245, intercambio de capacidades, mensajes y comandos de señalización para la realización adecuada en el terminal.
- Componentes de Transmisión: se encarga de los formatos transmitidos de audio, video, datos, tramas de control y mensajes para ponerlos en la interfaz de red, así como de su recepción.

- Codificadores de audio: codifican la señal de audio a transmitir y decodifican la que llega, para realizar esto son necesarias las funciones incluidas en el codificador/decodificador G.711 tales como velocidad y transmisión y recepción en los formatos *a-law* y *μ-law*. También pueden ser soportados los codificadores/decodificadores G.722, G.723.1, G.728 y G.729.
- Interfaz de red: un paquete basado en las capacidades de un interfaz capaz de realizar una transmisión punto a punto TCP y una transmisión unicast o multicast UDP.
- Canales de Datos: las aplicaciones soportan tanto el acceso a bases de datos, transferencias de archivos y conferencias.
- Codificadores de video: capaces de codificar y decodificar video según los estándar H.261/H.263.

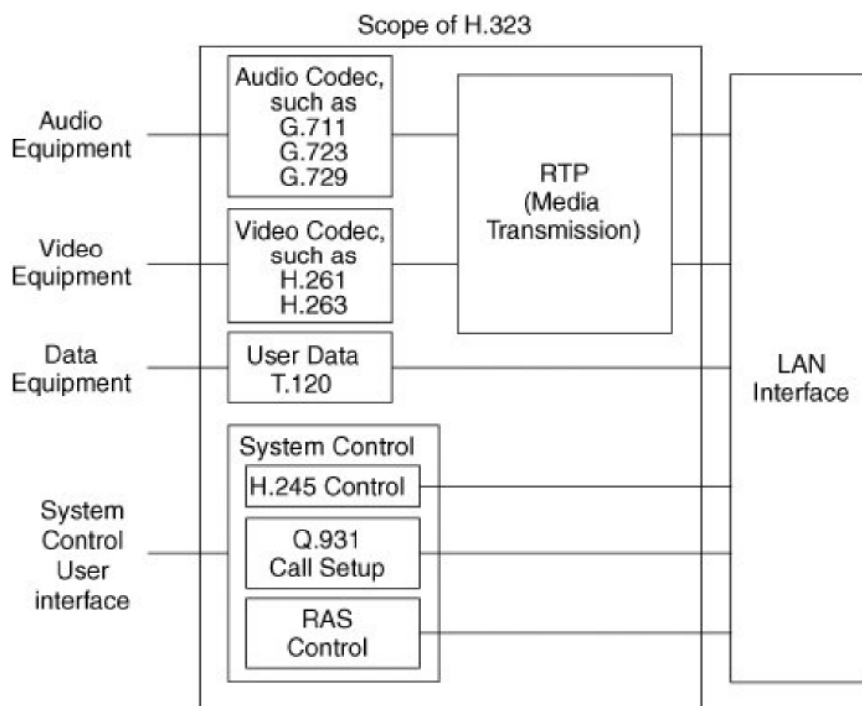


Fig.2.4 Imagen que muestra la relación que existe entre los componentes de H.323

SGCP

SGCP (*Simple Gateway Control Protocol*, Protocolo de Control de las Pasarelas) [SGCP] fue desarrollado en 1998 para disminuir el coste de los *gateways* creando un aparato que los controlara, pero fue superado por MGCP.

SGCP fue diseñado para que fuera compatible con SIP, daba la posibilidad al usuario de realizar llamadas entre una red de VoIP usando SIP y una red de telefonía tradicional.

IPDC

IPDC (*Internet Protocol Device Control*, Protocolo que Controla los Dispositivos de Internet) [IPDC] es similar a SGCP.

Es importante en el tema de transporte de la red porque trata el tema de desarrollo y mantenimiento de la red. Fue diseñado específicamente para controlar los

equipos de *hardware*, para que existiera una comunicación entre los *switches* centralizados y los *gateways*.

MGCP

MGCP es el primer protocolo desarrollado por IETF para señalar la información de control entre los componentes de redes VoIP. Se trata de SGCP con alguna aplicación añadida y es un predecesor incompleto de H.248.

Funciona como un conmutador tradicional de voz en tres unidades funcionales: el *gateway*, controlador de *gateways* y señalización *gateway*. Proporcionando al administrador de la red dirigir independientemente cada *gateway* de VoIP como una entidad separada.

MGCP es un protocolo de control *master-slave* que coordina las acciones de los *gateways*. El controlador de *gateway*, llamado *call agent*, dirige la información de la señal de control de la llamada, mientras que los *gateway* informan a los controladores del servicio de los eventos.

En la mayoría de los casos el *call agent* informa a los *gateways* para comenzar una sesión RTP entre los 2 puntos de la conexión.

SIP

SIP es el estándar que IETF utiliza para establecer las conexiones VoIP. Es un protocolo que pretende desarrollar los *gateways* para que sean más inteligentes.

Se trata de un protocolo de control de la capa de aplicación encargado de crear, modificar y terminar sesiones con uno o más participantes. Estas sesiones multimedia incluyen aplicaciones de audio, video y datos con varios participantes que se pueden comunicar a través de una comunicación *unicast*, *multicast* o una combinación de ambas.

Este protocolo de transporte usa invitaciones para crear mensajes *Session Description Protocol* (SDP, Protocolo que Describe la Sesión) [RFC4566] para llevar a cabo el intercambio y establecer el uso de canales de control.

La aplicación SIP cliente servidor, soporta usuarios móviles con dos modos de operación: *Proxy* y *Redirect*.

- Modo *Proxy*: cliente SIP envían petición al servidor *proxy*, el servidor *proxy* maneja las peticiones o las envía a otros servidores SIP. Los servidores *proxy* pueden aislar y esconder a los usuarios SIP del manejo de los mensajes de señalización. Para los otros usuarios de la red VoIP, la señal de invitación parece como si viniera del servidor *proxy* SIP.
- Modo *Redirect*: el cliente SIP envía al servidor la petición, el servidor busca la dirección de destino, que se la devuelve al cliente. Éste será quien envíe la señal al cliente SIP destino.

SIP es el protocolo básico para los *IP Multimedia Subsystem* (IMS, Subsistema Multimedia IP), datos multimedia y estructura del protocolo de control, que el IETF está desarrollando junto con *Third-Generation Partnership Project* (3GPP), aunque todavía no han llegado a una solución.

H.248

H.248 es el protocolo, tanto de IETF e ITU, usado para señalar la información de control entre los elementos de la red VoIP. Es el resultado de la colaboración entre IETF e ITU.

Define a los *gateways* que proporcionan conversiones entre enlaces y son fuentes de llamadas, mientras que los controladores de *gateways* proporcionan control de llamada. Define una serie de transacciones coordinadas por un controlador entre *gateways* para establecer la sesión de llamada.

Fue el principal promotor de la estandarización de equipo de telefonía IP e incluye:

- Teléfono IP que conoce las necesidades básicas del usuario de la empresa en un día.
- Un camino de rápida expansión para soportar características sofisticadas de telefonía de la empresa.
- Un ancho rango de teléfonos y aparatos similares para ser definidos de muy simples a muy ricas características.
- Mínimo diseño.
- El coste de los aparatos que son apropiados para la capacidad proporcionada y tipos de terminación con características que proporcionen fiabilidad.
- Un teléfono IP que conoce las necesidades del protocolo y que usa una simple aplicación de H.248.

2.7 Equipos de una red VoIP

2.7.1 Componentes de VoIP y equipo necesario

Una red VoIP tiene cinco componentes principales:

- Teléfonos VoIP, consolas, aplicaciones PC y otros aparatos para que cada usuario inicie y reciba llamadas VoIP
- El servidor que procesa las llamadas/PBX, quien maneja todas las conexiones de control de VoIP
- Uno o más *gateways* de PSTN a VoIP, convierten el contenido de voz para transportarlo sobre redes IP.
- La red IP, que transporta el audio que se transmite o recibe
- Uno o más *Session Border Controllers* (SBCs, Controlador de Sesiones), controlan el tiempo-real, el tráfico de la señalización en la llamada y la capa de transporte para limitar los cruces de redes y los dominios de redes

El equipo necesario para el establecimiento de una conexión de VoIP está compuesto por una conexión a *Internet* y para muchos tipos de llamadas VoIP, una conexión a *Internet* de banda ancha. Dependiendo del tipo de servicio VoIP que se adquiera, se necesitará un teléfono especial VoIP o un teléfono normal con un adaptador. Si se llama a un número de teléfono normal la otra persona no necesita ningún equipo especial, sólo un teléfono.

La Fig.2.5 nos muestra las formas de comunicarse en *Internet* y los aparatos que se necesitan para adaptar la señal analógica a la digital:

- Conexión de banda ancha (alta velocidad Internet) a través de MODEM, xDSL, LAN, etc...
- Ordenador, adaptador o teléfono especializado.
 - Con el ordenador se necesita software y un micrófono: un icono de teléfono aparece en la pantalla del ordenador, al pulsar el icono se permite marcar números o llamar mediante un programa de contactos nombre y número. Los ordenadores basados en los

- servicios VoIP tienen una variedad de maneras para notificar que se está realizando una llamada.
 - El teléfono especial VoIP se conecta directamente en la conexión de alta velocidad (*broadband*) y funciona como un teléfono tradicional
 - Teléfono con un adaptador VoIP, se marca el número de destino y el proveedor de servicios proporciona un tono de marcado
- *IP Softphones*: Es un teléfono virtual que se instala en un PC *Windows* o portátil. Los *IP softphones* se instalan en los ordenadores personales y contienen *software* para permitirles realizar llamadas con clientes de telefonía IP. Los PC contienen *speakers* y micrófonos que pueden operar similarmente a un teléfono de mano. Además, el *software* de telefonía IP digitaliza la señal de voz y envía los paquetes de voz a través de la red IP. Los *softphones* proporcionan un rico entorno para el desarrollo de aplicaciones TAPI (*Telephony Application Programming Interface*, Interfaz que Programa la Aplicación de Telefonía), como *Web click-to-talk* (pulsar para hablar) [CISC02].
- *Gateways*: Proporcionan un interfaz entre la red de telefonía IP y la PSTN. Son necesarios para permitir llamadas entre las localizaciones de VoIP y fuera de la red o localizaciones de PSTN. Las llamadas se realizan desde tu teléfono IP de tu oficina hasta un tradicional teléfono analógico y viceversa pasando a través de *gateway*. Proporcionan redundancia, cuando la red VoIP está congestionada o cuando la WAN (*Wide Area Network*, Red de Área Amplia) que lleva el tráfico VoIP está caída, el *gateway* desvía tus llamadas salientes de la WAN a la PSTN. El *gateway* que llama convierte los paquetes de voz digitales a una tradicional multiplexación por división en el tiempo (TDM, *Time Division Multiplexing*) de una trama de voz y transmite la llamada a través de las PSTN. El *gateway* destino convierte la trama de voz TDM que le llega en paquetes digitales para procesarlos por el teléfono IP de destino.
- *Switches*: desempeñan un papel importante en la transmisión de la llamada para teléfonos IP.

Otros de los equipos que podrían formar parte en nuestra red IP, porque las empresas los están demandando en aplicaciones multimedia, son: fax y videoconferencia en los PC.

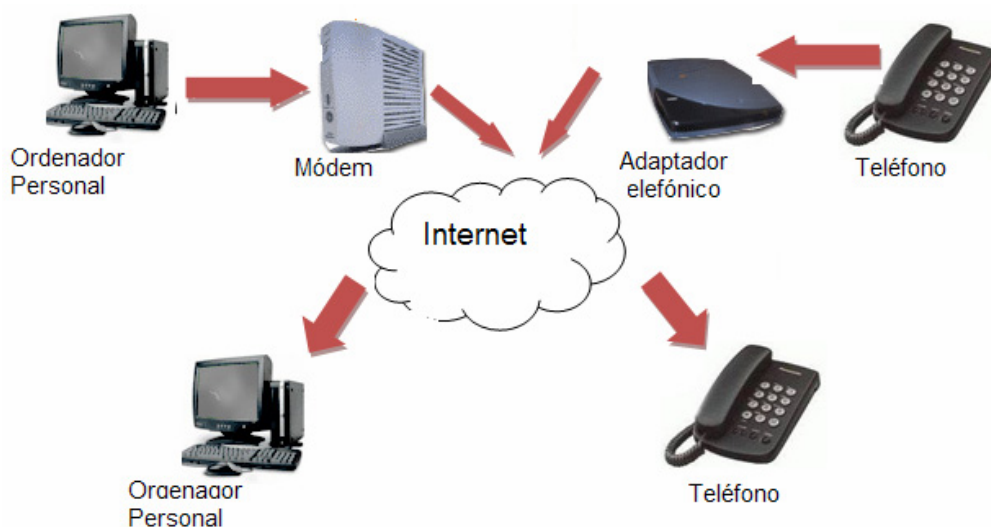


Fig.2.5 Muestra los dispositivos necesarios para transformar la señal analógica a digital

Para mayor información acerca de la creación de una red VoIP existe un proyecto dirigido por Cristina López Bravo, titulado “Estudio y desarrollo de soluciones VoIP para la conexión telefónica entre sedes internacionales de una misma empresa” [PFC1836]. En este trabajo se hace una descripción detallada de todos los dispositivos y protocolos necesarios para la creación de una red de VoIP.

2.7.2 Papel desempeñado por el FCC en VoIP

La FCC (*Federal Communications Commission*, Comisión Federal de Comunicaciones) ha trabajado para crear un medio promotor de competición e innovación para los beneficios de los clientes donde sea necesaria su actuación, para asegurar que los proveedores de VoIP cumplen con los requerimientos de seguridad.

En mayo de 2005 FCC impuso la obligación a los proveedores de habilitar el acceso a los servicios de emergencia (EE.UU. 911) que se realiza de forma diferente a las llamadas mediante teléfonos tradicionales.

FCC también limita la interconexión de proveedores VoIP usando la información de las redes propiedad de los clientes (CPNI, *Customer proprietary network information*), impide que los proveedores realicen uso de las redes de sus clientes para realizar acciones como pueden ser las grabaciones de sus llamadas telefónicas.

Finalmente, la interconexión de los proveedores VoIP debe cumplir con los requisitos de la *Commission's Section 225 Telecommunications Relay Services* (TRS, Servicio de Retransmisión en las Telecomunicaciones), incluyendo la contribución a *TRS Fund* usada para soportar la provisión de los servicios de telecomunicaciones a personas con disfunciones de habla u oído, y ofrece la posibilidad de realizar una marcación al código 711 para acceder a los servicios de TRS. TRS permite a las personas con problemas para hablar o escuchar usar el sistema telefónico a través de un teléfono con teclas para escribir texto u otros aparatos que les permita llamar a personas que tengan o no estas discapacidades físicas. Los proveedores deben asegurar que los servicios son disponibles y útiles para los individuos con discapacidades y que el acceso sea realmente eficaz.

2.7.3 Ejemplo de una llamada en VoIP

A continuación, realizamos un ejemplo de cómo se produciría una comunicación de VoIP entre dos personas, la Fig.2.6 muestra como se produce la negociación para que la llamada VoIP se produzca.

Tanto Juan como María necesitan estar en *Internet* o tener alguna otra red IP entre sus domicilios para poder comunicarse. Asumiendo que esta red existe entre ellos, veamos el flujo de llamada que se realiza:

1. María conecta su aplicación telefónica de *Internet*, que es compatible con H.323.
2. Juan ya tiene conectada su aplicación *phone*.
3. María conoce el “nombre” de *Internet* de Juan o su entrada DNS, así que ella pone “*who to call*” sección en su aplicación *phone* y presiona *Return*.
4. La aplicación *phone* convierte el DNS de Juan a un nombre de host de DNS y va a un servidor DNS (que estáticamente está configurado en la máquina de María) para resolver los nombres DNS y dar una dirección IP actual.
5. La máquina DNS le devuelve la dirección IP de Juan.

6. La aplicación de María toma la dirección IP de Juan y le envía un mensaje H.225 a Juan.
7. La señal del mensaje H.225 empieza a sonar en el PC de Juan.
8. Juan hace clic en el botón *Accept*, que dice su aplicación *phone* y devuelve un mensaje de conexión H.225.
9. La aplicación *phone* de María empieza una negociación H.245 con el PC de Juan.
10. La negociación H.245 finaliza y canales lógicos son abiertos. Por lo tanto Juan y María pueden hablar a través de una red basada en paquetes.

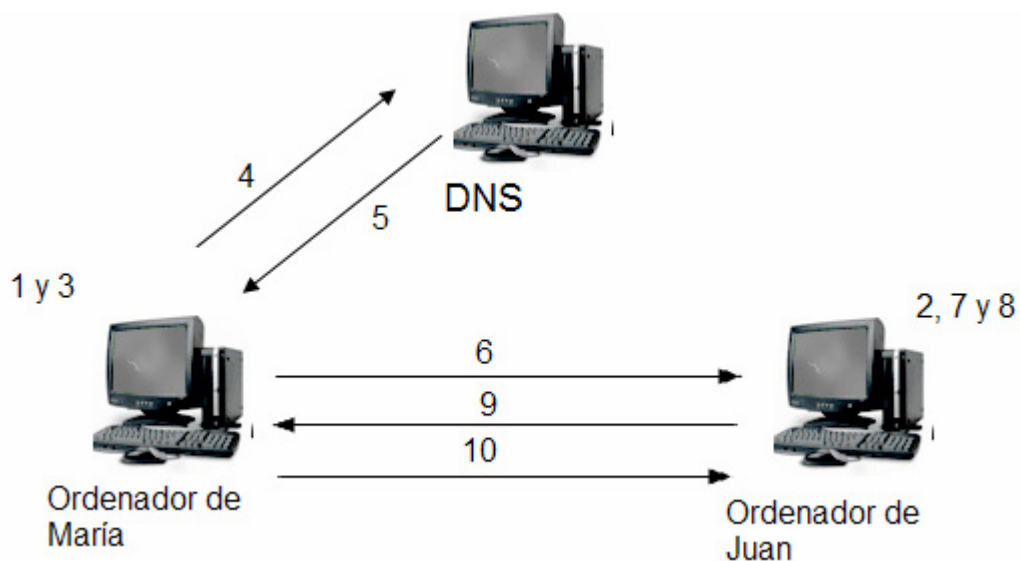


Fig.2.6 Comunicación de VoIP entre María y Juan

2.8 Aplicaciones de VoIP

Existen multitud de aplicaciones que ofrecen servicios de voz sobre IP, con distintas características que hacen que podamos elegir y valorarlas según nuestras necesidades.

Para conseguir el objetivo de este proyecto se clasificaron las distintas aplicaciones de VoIP que existen siguiendo estas características:

- Sistema Operativo: se trata del sistema operativo donde las aplicaciones pueden funcionar ya sea Windows, Linux, MAC, en teléfonos móviles, PDA's,...
- Tipo de código: se dividen en código abierto o cerrado, las de código abierto tienen la facilidad que el usuario puede modificar las características ofrecidas por la aplicación para potenciarlas a sus necesidades.
- Tipo de licencia: las hay comerciales, gratuitos, versión demo, licencia GNU, licencia GPL,...
- QoE ofrecida: se tiene en cuenta la valoración que el usuario cree que se merece tener la aplicación.
- QoS ofrecida

- Codificadores utilizados
- Protocolos utilizados
- Cifrado de los datos en la transmisión para habilitar una comunicación más segura.
- Características ofrecidas por la aplicación.
- Fecha de la última versión

Para poder medir la calidad de servicio ofrecida por este tipo de aplicaciones se eligieron cinco de ellas. En concreto, las que ofrecían mejor calidad experimentada por los usuarios: Gizmo5, Skype, Damaka, ooVoo y vBuzzer.

A continuación vamos a describir las características de cada aplicación VoIP.

Gizmo5

Se trata de una aplicación que se puede instalar en Linux, Mac, Windows, Symbian y teléfonos móviles. Su código se caracteriza porque es cerrado y su software gratuito.

La calificación puesta por los usuarios es un 9 sobre.

Los protocolos que usa son SIP, XMPP y Jabber, y la comunicación está cifrada mediante SRTP.

Sus principales características son que permite la realización de conferencia, videoconferencia, multiconferencia, correo de voz, mensajería instantánea, es posible llamara a teléfonos convencionales o móviles y recibir llamadas de ellos, se pueden realizar transferencias de archivos y también está capacitado para enviar SMS.

Su última versión es Gizmo5 4.0.1.344 creada en marzo de 2008 [GIZ08] (véase Fig.2.7).



Fig.2.7 Imagen de la aplicación Gizmo5

Skype

Se puede instalar en múltiples plataformas como pueden ser Windows, teléfonos móviles con WiFi, Mac, Linux y también para teléfonos inalámbricos. Se trata de un código cerrado pero que está programado en lenguaje Pascal y su software es libre.

La calificación realizada por los usuarios es un 8.6 sobre 10.

Tanto los protocolos que usa como los codificadores son de código propietario. Emplea AES y RSA para cifrado.

Sus principales características son que con esta aplicación se pueden realizar conferencias, videoconferencias, transferencia de archivos, correo de voz, mensajería instantánea, se pueden realizar llamadas a teléfonos convencionales y también se pueden recibir llamadas de ellos. Como extensiones adicionales tienen la posibilidad de realizar juegos P2P o usar la pizarra compartida para comunicarse con los contactos.

Su última versión para Windows es Skype 3.6.0.248 creada en febrero de 2008 [SKY08] (véase Fig.2.8).

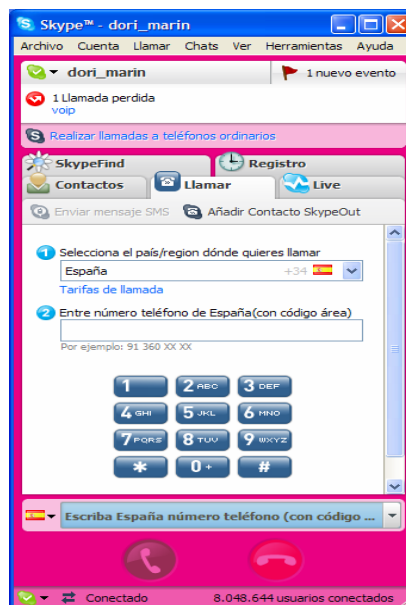


Fig.2.8 Imagen de la aplicación Skype

Damaka

Su software gratuito y de código cerrado se puede instalar en el sistema operativo Windows. Está valorado por los usuarios con una nota de 7.6 sobre 10.

Los protocolos que usa son SIP, P2P, H.263, H.264. El cifrado se realiza punto a punto.

Las características ofrecidas por esta aplicación son videoconferencia, transferencia segura de archivos, escritorio compartido, correo de voz, posibilidad de enviar SMS, radio IP, mensajería instantánea y pizarra compartida.

Su última versión es Damaka 3.1.203 y fue creada en diciembre de 2007 [DAM07] (véase Fig.2.9).



Fig.2.9 Imagen de la aplicación Damaka

OoVoo

Es una aplicación que se puede instalar en sistemas operativos Windows y Mac OS X, su código es cerrado y su software es gratuito. La calificación con la que los usuarios lo han puntuado es de 9 sobre 10.

Los protocolos que usa son de código propietario basado en On2 TrueMotion VP7.

Las principales características de ooVoo son que es capaz de realizar videoconferencia puede soportar hasta 6 personas simultáneamente, se pueden crear mensajes de video, mensajería instantánea y es posible transferir archivos.

Su última versión es la ooVoo 1.5.1.97 en abril de 2008 [OOV08] (véase Fig.2.10).



Fig.2.10 Imagen de la aplicación ooVoo

vBuzzer

Es una aplicación de código cerrado y el software es gratuito. Se puede instalar en los sistemas operativos Windows.

La nota con la que han calificado los usuarios la aplicación es un 7 sobre 10.

El protocolo que usa es SIP y el cifrado es TLS.

Las características de este software son conferencia, mensajería instantánea, se pueden crear correos de voz y realizar transferencias de ficheros.

La última versión es vBuzzer 2.5.054 creada en enero de 2008 [VBU08].

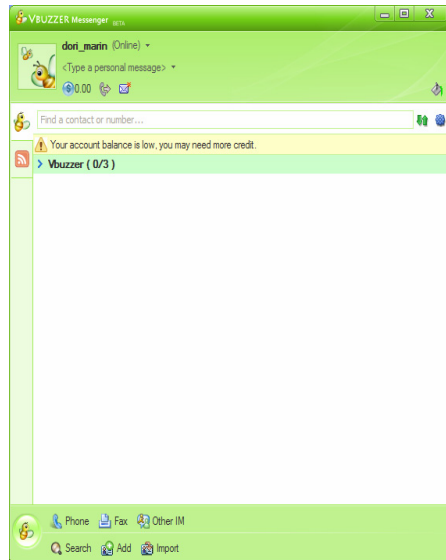


Fig.2.11 Imagen de la aplicación vBuzzer

Las aplicaciones están calificadas por los usuarios como las mejores aplicaciones de VoIP que existen hoy en día. Con ellas realizaremos un estudio experimental de la calidad de servicio ofrecida y la compararemos con la calidad de servicio experimentada por los usuarios.

Capítulo 3

Análisis de QoS en aplicaciones VoIP de libre distribución

3.1 Introducción

En este capítulo nos centraremos en medir la calidad de servicio ofrecida por las aplicaciones de voz sobre IP de libre distribución elegidas. Estas herramientas son Gizmo, Skype, Damaka, ooVoo y vBuzzer.

Primeramente enumeraremos los parámetros que hemos considerado más importantes para cuantificar la calidad de servicio de estas aplicaciones. A continuación explicaremos el ámbito en el que se han medido estas aplicaciones y las herramientas usadas para ello. Por último, vislumbraremos todas las gráficas halladas para este estudio, comentando sus valores y realizando una conclusión de este análisis.

3.2 Parámetros usados para el análisis

La calidad de servicio es un parámetro que en los inicios de las redes IP fue ignorado pero que hoy en día es imprescindible. Cada día la gente es más dependiente de las redes IP debido a la gran cantidad de servicios y aplicaciones que ofrecen. Y cada día las exigencias para estos recursos aumentan, se intenta alcanzar la perfección en las comunicaciones que establece.

Por lo tanto, si queremos que en las comunicaciones de voz IP la calidad se parezca a la ofrecida por un canal dedicado exclusivamente para este uso necesita ir generando nuevas aplicaciones con más prestaciones. Cuando se trata de progreso para VoIP el mayor reto es garantizar y aumentar la calidad de servicio.

Para poder medir la calidad de servicio ofrecida por las aplicaciones seleccionadas (Gizmo5, Damaka, Skype, ooVoo y vBuzzer) realizamos capturas de conversaciones con la herramienta *Wireshark*. A continuación procedimos a guardar esa captura en texto claro para que pudiera ser interpretada por un script awk que habíamos creado. Las mediciones fueron hechas para obtener los siguientes parámetros:

- Ancho de banda
- Jitter
- Tiempo entre paquetes
- Paquetes perdidos

3.2.1 Ancho de banda

Como ya establecimos en el capítulo anterior el ancho de banda disponible en una red es uno de los parámetros más apreciados para la VoIP. En las redes IP existen multitud de aplicaciones ejecutándose simultáneamente, tanto de datos como de video y voz y necesitamos que estos últimos se transmitan sin problemas.

Existe gran diferencia entre los paquetes de datos y los que son de video y voz, debido a que los últimos son excesivamente sensibles al retraso y al colapso que se pueda producir en la red. Por tanto se tiene que garantizar un ancho de banda disponible para este tipo de paquetes, para ello se aplican distintos algoritmos que les apliquen prioridades.

Podemos establecer una relación directamente proporcional entre la cantidad de ancho y la calidad de los datos transmitidos. No obstante, necesitamos aplicaciones que no consuman un ancho de banda excesivo de lo contrario estarían requiriendo un uso exclusivo del canal para su transferencia, pero necesitamos que sí tengan el necesario para su transmisión con éxito.

3.2.2 Jitter

Este es un factor que afecta al retraso y a la pérdida de paquetes. Sólo existe en las redes basadas en paquetes y se define como la variación del tiempo entre paquetes con el que el transmisor los envía a la red y el receptor los recibe.

El jitter se produce normalmente en redes que tienen un ancho de banda de baja capacidad. Puede causar que los paquetes a su llegada al receptor sean procesados fuera de secuencia. Las aplicaciones de voz son muy susceptibles a estas variaciones ya que es un parámetro que degrada cuantiosamente la calidad de la voz y modifica el diálogo. Lo que influye directamente en la QoS de la VoIP.

3.2.3 Tiempo entre paquetes

El tiempo que debe existir en la recepción entre los paquetes de voz tiene que ser un valor bajo ya que si fuera alto el oyente percibiría que la conversación no es fluida y se entrecortaría debido al retraso que se provocaría de un paquete a otro.

Normalmente los paquetes de voz IP son enviados cada 20 ms, esto garantiza que la calidad de voz percibida por el receptor no sufre modificaciones debido a la diferencia de llegada de un paquete a otro. Esta diferencia tiene que ser constante y mantener una diferencia que el oído humano no note que se están transmitiendo en paquetes distintos.

3.2.4 Paquetes perdidos.

La pérdida de paquetes puede ocurrir por diversas circunstancias, las principales son congestión en la red y excesivo retardo. La congestión de la red es debida al tamaño de las colas, pueden descartar paquetes porque estén llenas. La pérdida de paquetes también puede ser resultado de un excesivo retardo, donde un grupo de paquetes llegan tarde y deben ser descartados en favor de los nuevos.

El principal problema de la voz IP es que las aplicaciones están basadas en tiempo real y usan el protocolo UDP. Este protocolo de transporte es poco tolerante a la pérdida de paquetes ya que no tiene posibilidad de retransmisión.

La longitud de los paquetes destinados a VoIP es pequeña, esto supone que la pérdida de uno de ellos no supone una disminución de la calidad en la conversación. Sin embargo, probabilísticamente la pérdida de un paquete significa la pérdida de varios paquetes.

Como vimos en el capítulo anterior la calidad de servicio ofrecida por una de estas aplicaciones no se ve disminuida con un porcentaje de pérdidas inferior al 3%, más de esto supone una gran degradación en la calidad de la conversación.

3.3 Descripción de las mediciones a tomar

Para poder establecer unos valores óptimos en el estudio de la calidad de servicio se realizaron capturas de las llamadas. Las capturas fueron realizadas mediante la herramienta Wireshark.

El estudio ha consistido en la realización de llamadas entre dos puntos que no pertenecieran a la misma red IP y que estuvieran separados por varios *routers*. Cada llamada tenía como duración mínima cinco minutos, caracterizada porque siempre se leía el mismo texto [CLU04]. Las llamadas se produjeron en tres franjas horarias: mañana, tarde y noche. Con cada aplicación se realizaron cinco capturas en cada uno de los distintos horarios. Las capturas se iniciaban y terminaban en el mismo instante temporal, de esta manera las posteriores mediciones realizadas son más exactas. Mencionar también que las llamadas se produjeron en un entorno silencioso, ya que no existían ruidos de ningún tipo, ni provenientes de personas, ni de maquinaria.

También hay que mencionar que el estado de la conexión de ambos extremos era de calidad excelente. Además mientras se realizaban las capturas ninguno de los extremos tenía un uso del canal para otra aplicación, de este modo la aplicación de voz IP no tenía que competir por el canal para establecer una conversación. Las condiciones para video eran de luz blanca e indirecta a la cámara *web*, de este modo no se producía reflejo.

Durante el establecimiento de la conexión entre transmisor y receptor se mantuvo habilitada la *Web Cam* sólo en aquellas aplicaciones que lo permitían (vBuzzer era la única que no permitía el uso de cámara).

3.4 Herramientas de análisis.

Las herramientas utilizadas para poder hallar los parámetros necesarios fueron Wireshark y awk.

3.4.1 Wireshark.

Las capturas se realizaron con la herramienta Wireshark (véase Fig.3.1) cada columna está compuesta por unos campos que seleccionamos entre los ofrecidos por esta herramienta. Se eligieron los campos que nos daban información sobre el tiempo entre paquetes, el tiempo en el que llegaban los paquetes dentro de la captura realizada y la cantidad de bytes que contenía cada paquete.

Después de realizar la captura se aplicaba un filtro. Para las capturas realizadas tanto por el receptor como por el transmisor el filtro consistía en seleccionar los paquetes que tenían como dirección IP origen la del transmisor y como destino la del receptor. De este modo sólo nos quedaban los paquetes que se habían producido en la conversación.

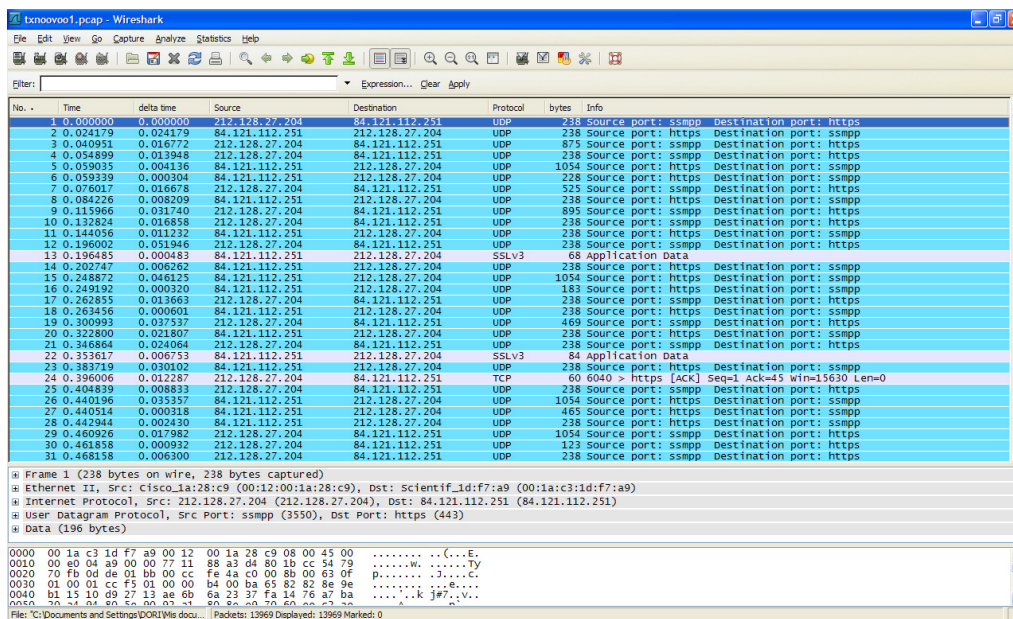


Figura 3.1 Captura realizada con la aplicación Wireshark

3.4.2 Awk.

Después de aplicarles un filtro a las capturas realizadas con Wireshark se les aplicó un *script* en *awk* para conseguir medir distintos parámetros. En concreto, se realizaron dos *script*: el primero se aplicaba a las capturas realizadas por el transmisor y el segundo a las capturadas por el receptor. Existe una correspondencia entre las columnas de la aplicación *Wireshark* y las variables que aparecen en el *script*:

- \$2 se corresponde con la segunda columna de Wireshark e indica el tiempo en el que ha sido recibido ese paquete.
- \$3 se corresponde con la tercera columna de Wireshark e indica el tiempo en el que ha sido recibido ese paquete de VoIP con respecto al anterior.
- \$8 se corresponde con la tercera columna de Wireshark e indica el número de *bytes* que tiene ese paquete.

El siguiente *script* (véase Fig.3.2) es el aplicado a las capturas realizadas en el transmisor. Conforme se va recorriendo el fichero se van actualizando las variables que nos interesan. Por cada línea leída vamos incrementando el contador de número de paquetes leídos, se va calculando el tiempo que existe entre un paquete y el anterior que se ha transmitido, se calcula el número total de *bits* que se transmiten y se calcula en ancho de banda que ha usado cada paquete en su transmisión. Tanto para el tiempo entre paquetes como para el ancho de banda se calculan su media y su varianza. Cuando se ha finalizado de leer el fichero se imprimen los parámetros en un archivo creado para este fin.

Para la captura realizada en el receptor (véase Fig.3.3) se realizó otro *script* que tenía todo lo que el anterior más otras características añadidas. Al igual que el anterior, se encarga de escribir en el archivo en el que se están direccionando los resultados. Además realiza la lectura de las filas de ese mismo fichero que contienen los valores numéricos y realiza diversas mediciones como son el cálculo de los paquetes perdidos, la media del jitter y la varianza del jitter. A continuación de realizar

los cálculos los escribe en el mismo fichero. De este modo viendo un solo fichero obtenemos todos los valores que nos interesan.

```

#!/bin/awk -f
#Primero comprobaremos el tiempo entre paquetes

{
    count_packet++
    t_packet+=$2-t_paq_anterior
    t_packet_cua+=$(2-t_paq_anterior)^2

#Ancho de banda que ocupa cada paquete
    total_bits+=$8*8

    BW+=$8*8/$3
    BW_cua+=$(8*8/$3)^2
    t_paq_anterior=$2
}

END{
#media y varianza del tiempo que existe entre paquetes
    m_t_packet=t_packet/count_packet
    v_t_packet=t_packet_cua/count_packet-(m_t_packet)^2

#media y varianza del ancho de banda
    m_BW=BW/count_packet
    v_BW=BW_cua/count_packet-(m_BW)^2

#impresion por pantalla
{format = "%-10s %-13s %-13s %-8s %-11s \n"
print "Transmitidos por la fuente \n" >> "no1.txt"
printf format, "N°Paquetes", "tm_entre_paq", "tv_entre_paq", "media_BW",
"varianza_BW" >>"no1.txt"
printf format, "-----", "-----", "-----", "-----",
"-----" >>"no1.txt"}
{printf format, count_packet, m_t_packet, v_t_packet, m_BW, v_BW
"\n">>"no1.txt"}
}

```

Figura 3.2 Script en awk realizado para la captura en el transmisor.

```

#!/bin/awk -f
#Primero comprobaremos el tiempo entre paquetes
{
    count_packet++
    t_packet+=$2-t_paq_anterior
    t_packet_cua+=$(2-t_paq_anterior)^2

#Ancho de banda que ocupa cada paquete
    total_bits+=$8*8

    BW+=$8*8/$3
    BW_cua+=$(8*8/$3)^2
    t_paq_anterior=$2
}
END{
#media y varianza del tiempo que existe entre paquetes
    m_t_packet=t_packet/count_packet
    v_t_packet=t_packet_cua/count_packet-(m_t_packet)^2

#media y varianza del ancho de banda
    m_BW=BW/count_packet
    v_BW=BW_cua/count_packet-(m_BW)^2

#impresion por pantalla
{
format = "%-10s %-13s %-13s %-8s %-11s \n"
print "Recibidos en destino procedentes de la fuente \n" >>
"no1.txt"
printf format, "N°Paquetes", "tm_entre_paq", "tv_entre_paq",
"media_BW", "varianza_BW" >>"no1.txt"
printf format, "-----", "-----", "-----", "-----",
"-----", "-----" >>"no1.txt"
printf format, count_packet, m_t_packet, v_t_packet, m_BW, v_BW
"\n" >>"no1.txt"
}
close("no1.txt")

#Recorremos el fichero para calcular el jitter en media y
varianza y los paquetes perdidos
while((getline <"no1.txt">0){
if($1~"[0-9]{9}") {
        paq_perdidos=$1-paq_perdidos
        jitter_m=$2-jitter_m
        jitter_v=$3-jitter_v
    }
}
close ("no1.txt")
{
formato= "%-12s %-12s %-15s \n"
printf formato, "Paq_perdidos", "Jitter_media",
"Jitter_varianza">>"no1.txt"
printf formato, "-----", "-----", "-----"
>>"no1.txt"
printf formato, paq_perdidos, jitter_m, jitter_v >>"no1.txt"
}
}
}

```

Figura 3.3 Script en awk realizado para la captura en el receptor.

A continuación mostramos un ejemplo del archivo generado (véase Fig.3.4) con los script que posee todos los parámetros buscados para poder determinar la calidad de servicio.

Transmitidos por la fuente			
N°Paquetes	tm_entre_paq(seg)	tv_entre_paq(seg)	media_BW(bps)
-----	-----	-----	-----
4348	0,0241585	0,000331641	56418,7
Recibidos en destino procedentes de la fuente			
N°Paquetes	tm_entre_paq(seg)	tv_entre_paq(seg)	media_BW(bps)
-----	-----	-----	-----
4141	0,0599955	0,000460466	1,32873e+06
Paq_perdidos	Jitter_media(seg)	Jitter_varianza(seg)	
-----	-----	-----	
-207	0,035837	0,000128825	

Figura 3.4 Archivo generado con los script en awk y las capturas realizadas.

3.5 Resultados y discusión.

La principal característica de las llamadas es que al ser aplicaciones de voz IP la transmisión se producía mediante el protocolo de transporte UDP. Sin embargo en la aplicación Gizmo, en tramos horarios en los que se producía más colapso de las redes, usaba el protocolo de transporte TCP para conectarse con un servidor Gizmo. Mediante este se realizaba la conexión al otro extremo de la comunicación.

Otra característica de las llamadas es que eran muy estables, no se realizó la desconexión inesperada de ninguna de las aplicaciones. Sin embargo, con la aplicación Damaka si se tuvo problemas para identificar el estado de sus contactos y con la aplicación ooVoo también se tuvo problemas ya que tenía fallos al autenticar nombre y contraseña en su servidor.

Tenemos que mencionar que las aplicaciones tenían problemas para usar conjuntamente voz y video, por ejemplo Gizmo no los podía ofrecer conjuntamente durante la mañana y la noche. Además Skype tenía problemas con la definición del video durante la noche y por la mañana no se podían realizar conjuntamente la comunicación. También tenemos que recordar que la aplicación vBuzzer carece de video, por lo que los resultados se ven influenciados por este hecho.

A continuación vamos a observar las gráficas de los parámetros elegidos anteriormente que nos muestran los resultados obtenidos para cada tramo horario y para cada aplicación.

3.5.1 Tiempo medio entre paquetes.

En las siguientes gráficas podemos ver el tiempo medio que existe entre dos paquetes consecutivos de voz, para cada una de las cinco aplicaciones en estudio.

Capturas Mañana

En las capturas realizadas por la mañana (véase Fig.3.5) podemos observar dos grupos bien diferenciados, uno compuesto por las aplicaciones Gizmo y ooVoo, que poseen altos valores de tiempo medio entre paquetes, y el otro formado por Skype, Damaka y vBuzzer, cuyos tiempos son más bajos.

- Para la aplicación Gizmo este tiempo se encuentra entre los valores de 60 y 70 ms.
- Para aplicación Skype el tiempo entre paquetes se mantuvo constante en las cinco capturas realizadas obteniendo un valor de 17 ms.
- Para Damaka las tres primeras capturas tuvieron un valor constante de 20 ms, no obstante en las dos capturas restantes disminuye hasta 10 ms.
- El tiempo entre paquetes de voz IP para la aplicación ooVoo disminuye un poco con respecto al valor hallado para Gizmo, aunque es más constante y de menor valor que ésta. Se mantiene entre los valores de 55 y 60 ms.
- Para vBuzzer este tiempo se mantiene constante en las cuatro primeras capturas con un valor aproximado de 20 ms, pero en la quinta se incrementa un poco este valor llegando a 30 ms. Al contrario que para Damaka que disminuía, ésta se incrementa.

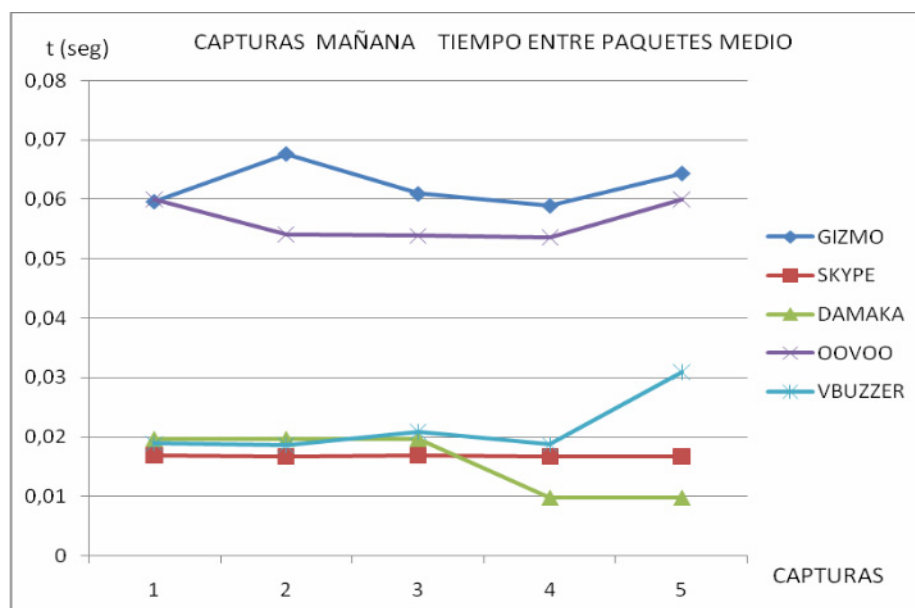


Figura 3.5 Gráfica del tiempo medio entre paquetes para las capturas realizadas durante la mañana

Capturas Tarde

En las capturas realizadas por la tarde (véase Fig.3.6) vemos que todas las aplicaciones se mantienen constantes para las cinco capturas. También observamos que podemos encontrar tres grupos, uno formado por la aplicación Gizmo, tiene los valores más altos, otro formado por la aplicación ooVoo y el tercer grupo se forma de las tres aplicaciones restantes, con los valores más bajos. Comparando con las

capturas de la mañana observamos cómo se mantienen el orden en los resultados de las aplicaciones.

- En la aplicación Gizmo vemos que sus valores se encuentran entre los 90 y 100 ms, vemos que ha crecido considerablemente el tiempo medio que existe entre paquetes con respecto a las capturas realizadas por la mañana.
- La aplicación Skype es la que nos ofrece un menor tiempo medio entre paquetes durante la tarde. Sus valores se encuentran en torno a los 15 ms, siendo este valor aproximado al hallado durante la mañana.
- En Damaka este tiempo tiene un valor de 20 ms para las cinco capturas, su valor no discierne mucho de las capturas de la mañana, sin embargo en esta franja horaria las cinco capturas son constantes.
- En la aplicación ooVoo este tiempo se encuentra sobre los 60 ms, por lo tanto podemos observar que no ha variado con respecto a los resultados de la mañana. Sigue siendo un valor alto para este tipo de tiempo.
- En vBuzzer el tiempo medio entre paquetes se encuentra sobre los 25 ms aproximadamente, estos valores son similares a los que se producen en las capturas realizadas durante la mañana. No obstante los valores hallados son iguales para las cinco capturas, cosa que no sucedía en las capturas de la mañana.

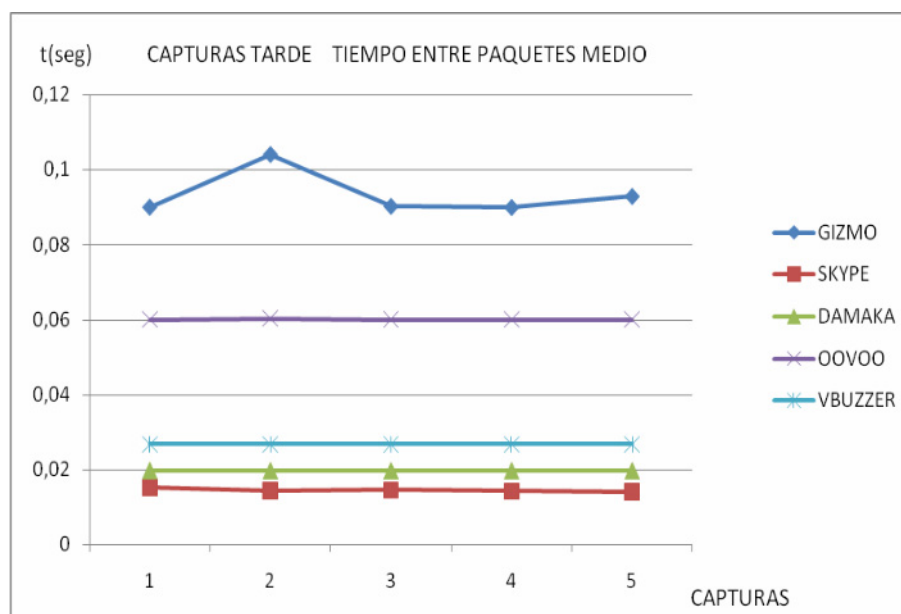


Figura 3.6 Gráfica del tiempo medio entre paquetes para las capturas realizadas durante la tarde

Capturas Noche

Los resultados hallados de las capturas realizadas durante la noche (véase Fig.3.7) nos muestran que los valores son aproximados a los obtenidos en las otras dos franjas horarias. Se siguen manteniendo Gizmo y ooVoo como las aplicaciones que ofrecen mayor tiempo entre paquetes de voz IP.

- En la aplicación Gizmo disminuye con respecto a las otras franjas temporales, pero su valor sigue siendo alto, entre 51 y 60 ms, mantiene constancia entre las cinco capturas.
- La aplicación Skype tiene unos valores comprendidos entre 5 y 20 ms, observamos que es más variable que en las otras dos franjas horarias. Sin embargo esta variabilidad entre capturas tiende a disminuir el tiempo medio entre paquetes.
- La aplicación Damaka también se mantiene constante a 20 ms al igual que en las capturas realizadas por la tarde.
- En ooVoo vemos que sus valores son más variables comprenden el intervalo entre 38 ms hasta 60 ms, aunque las últimas tres capturas se mantienen constantes al valor más alto.
- El programa vBuzzer tiene los mismos valores que nos proporcionaban las capturas de la tarde con un valor de 27 ms.

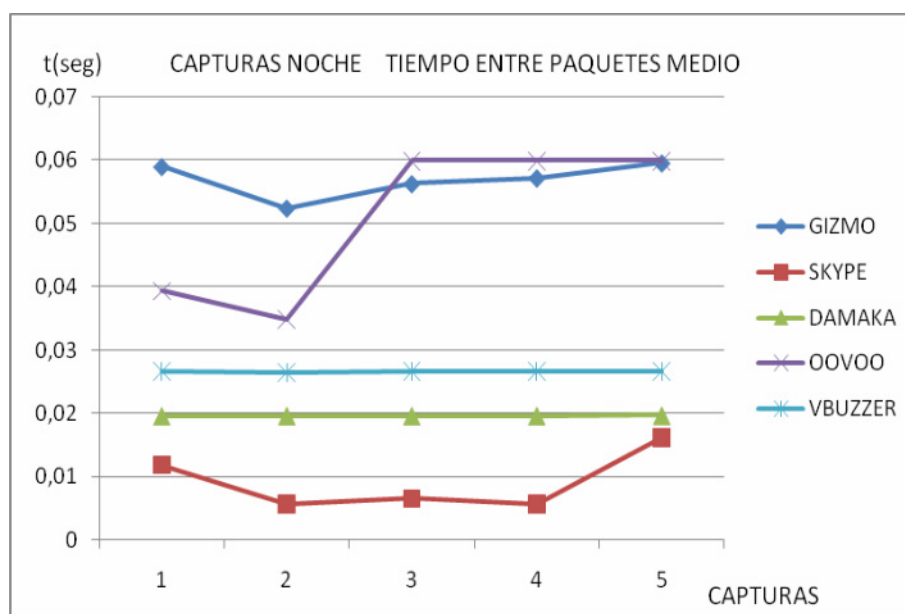


Figura 3.7 Gráfica del tiempo medio entre paquetes para las capturas realizadas durante la noche

Conclusión del tiempo medio entre paquetes

Podemos concluir que lo deseable es que el tiempo medio entre la transmisión de dos paquetes de voz consecutivos sea mínima y que sea inferior a 20 ms como recomiendan los estándares.

En los tres horarios el programa que ofrece más tiempo medio entre paquetes de voz para las cinco aplicaciones es el Gizmo, esto es una característica negativa de esta aplicación. Podemos apreciar que para la aplicación donde este tiempo difiere más de una captura a otra es también para Gizmo. Este hecho incrementa la negatividad de la aplicación ya que no se establece una constancia en la actividad. Además supera los 20 ms recomendables entre paquetes.

El ooVoo aunque en menor grado es semejante al Gizmo, siendo igual de negativo si tenemos en cuenta el tiempo medio entre paquetes de voz IP. La aplicación vBuzzer vemos que en las capturas realizadas durante la tarde y la noche son cuando los valores difieren menos de unas capturas a otras. Aunque su valor es ligeramente superior al deseado. Para la aplicación Damaka se produce la misma situación que para vBuzzer, no obstante sus valores son menores que los de la anterior aplicación y están dentro del margen buscado.

Para este factor podemos considerar que la mejor aplicación sería Skype debido a que es la que ofrece menor tiempo entre paquetes de voz IP. También es un requisito importante que para las capturas de la mañana y la tarde sus valores son constantes. En las de la noche, aunque los resultados varíen de unas capturas a otras nunca sobrepasan el máximo valor hallado en los otros dos tramos horarios. Incluso tienden a ser menores los valores del tiempo medio entre paquetes.

3.5.2 Varianza del tiempo entre paquetes.

En las siguientes gráficas podemos ver la varianza que existe del tiempo entre dos paquetes consecutivos de voz, para cada una de las cinco aplicaciones bajo estudio.

Capturas Mañana

Para las capturas realizadas durante la mañana (véase Fig.3.8) podemos apreciar que el programa Gizmo tiene valores de varianza del tiempo entre paquetes mayores comparados con el resto de aplicaciones. Aunque estos valores son mínimos, del orden de las diezmilésimas. Nos indica la diferencia que existe entre el tiempo de llegada de los paquetes.

- La aplicación Gizmo tiene un varianza del tiempo medido entre paquetes de voz IP muy variable comprendido entre valores de 6 y 9 diezmilésimas.
- Podemos apreciar que Skype es la aplicación cuya varianza es menor y tiene el aditivo que se mantiene constante para todas las capturas realizadas.
- La aplicación Damaka también observamos que proporciona pequeños valores de varianza sin embargo es ligeramente mayor que la de Skype. También se mantiene constante para todas las capturas realizadas.
- En varianza existente en las capturas realizadas a ooVoo podemos ver que está por debajo de Gizmo siendo de los más altos, sus valores se encuentran entre 3 y 5 diezmilésimas.
- Finalmente, la aplicación vBuzzer muestra diferencias entre unas capturas y otras, los valores que se obtienen están comprendidos entre 3,5 y 1 diezmilésima. Vemos que de unas capturas a otras difieren sus datos.

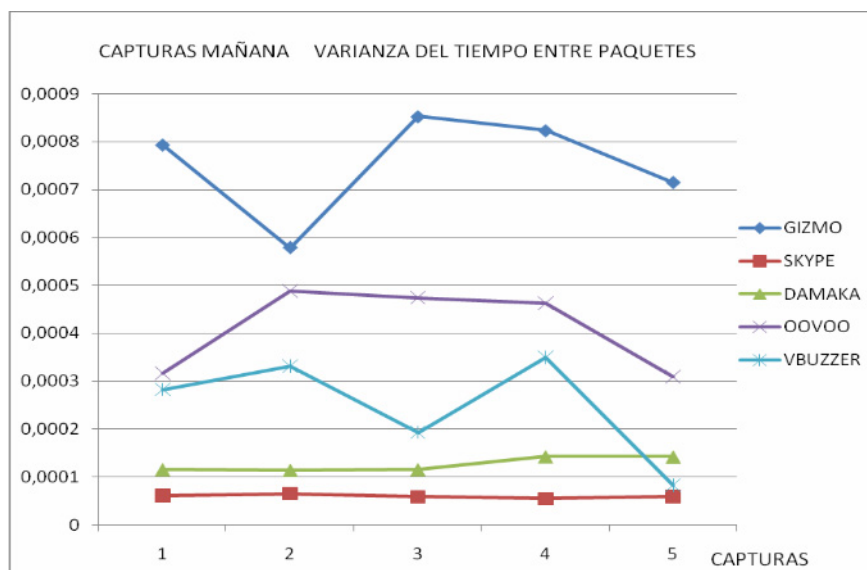


Figura 3.8 Gráfica de la varianza tiempo entre paquetes para las capturas realizadas durante la mañana

Capturas Tarde

En las capturas realizadas durante la tarde (véase Fig.3.9) vemos que existen diferencias entre las cinco aplicaciones. Con los valores más altos se encuentra Gizmo sin embargo ooVoo es la aplicación que más varía de una capturas a otras, las otras tres aplicaciones se mantienen aproximadamente en los mismos valores y similares en todas sus capturas.

- La aplicación Gizmo tiene su mínimo en 11 diezmilésimas y su máximo en 13,5. Es superado por una de las capturas de ooVoo, pero en general podríamos decir que Gizmo es el que proporciona los mayores valores constantes para todas las capturas realizadas. Por la tarde ofrece mayores valores de varianza que durante la mañana.
- La aplicación Skype tiene el mismo valor para todas las capturas, está caracterizado por la constancia de su aplicación. Sin embargo su valor es ligeramente superior al obtenido en las capturas matinales.
- La que menor variación ofrece es Damaka. Esta aplicación tiene los mismos valores que los determinados durante la mañana y es aproximadamente iguales para todas las capturas valiendo 1,3 diezmilésimas.
- La aplicación ooVoo es la que posee los valores más variados se encuentran entre 2 y 15 diezmilésimas, alcanzando el máximo de todas las aplicaciones. Este margen es seis veces superior al obtenido en las capturas de la mañana. Esto demuestra que no es constante en todas las capturas que es muy variable y esta aplicación no proporciona herramientas para solucionar este problema.

La aplicación vBuzzer se mantiene más constante para todas las capturas que durante la mañana. Su valor es aproximadamente 2,3 diezmilésimas para todas las aplicaciones.

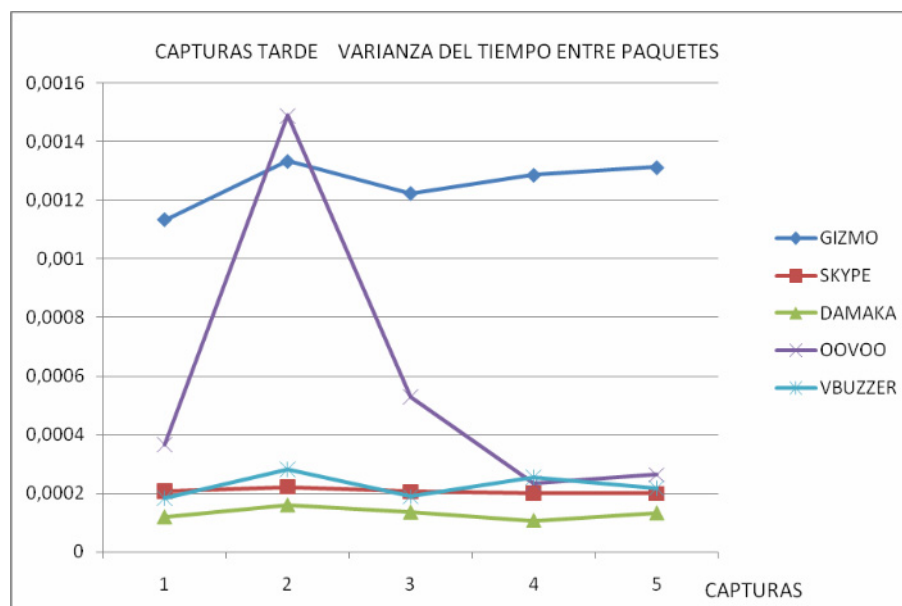


Figura 3.9 Gráfica de la varianza del tiempo entre paquetes para las capturas realizadas durante la tarde

Capturas Noche

En las capturas realizadas durante la noche (véase Fig.3.10) vemos, que las fluctuaciones son mayores en los programas Gizmo y ooVoo, como en el resto de franjas horarias. En los otros tres programas, Skype, Damaka y vBuzzer, la varianza es muy pequeña en las cinco capturas, pero más variable y de mayores valores que los obtenidos tanto en la mañana como en la tarde.

- La aplicación Gizmo es la que adquiere mayores valores comprendidos entre 4 y 7 diezmilésimas. Estos valores son menores que los obtenidos durante la tarde, siendo aproximadamente la mitad de ellos y también son menores, pero en menor medida, que los de la mañana.
- En Skype la varianza disminuye con respecto a los resultados hallados durante el tramo de la tarde, son aproximadamente iguales a los de la mañana. Sin embargo, podemos apreciar que son más variables que estos. Constituyen los valores de varianza más bajos de entre las cinco aplicaciones estudiadas.
- La aplicación Damaka sigue proporcionando valores similares a los obtenidos tanto durante la mañana como durante la tarde, comprendidos entre 1 y 2 diezmilésimas.
- En la aplicación ooVoo los valores están comprendidos entre 4 y 9 diezmilésimas. Aunque el margen de valores es menor que los que se dieron durante la tarde es superior al de la mañana. Constituye la aplicación que alcanza mayores valores y que carece de homogeneidad entre las capturas.
- La aplicación vBuzzer tiene valores comprendidos entre 2 y 4,5 diezmilésimas. Esta margen es superior al obtenido durante la tarde y ligeramente inferior al de la mañana. Las tres primeras capturas tienen el

mismo valor, 2 diezmilésimas, que aumenta hasta 4.5 en las 2 últimas capturas realizadas.

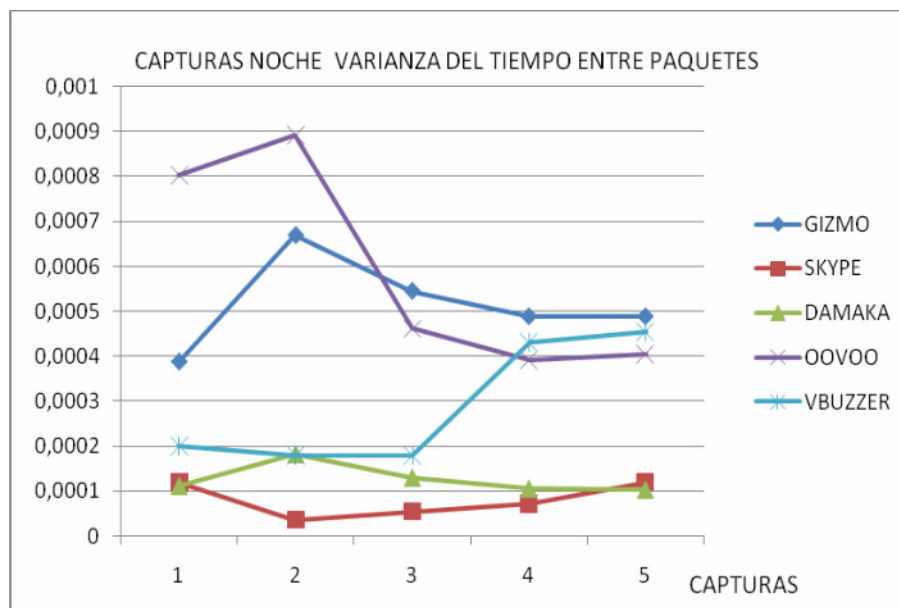


Figura 3.10 Gráfica de la varianza del tiempo entre paquetes para las capturas realizadas durante la noche

Conclusión de la varianza del tiempo entre paquetes

Hemos observado que la aplicación Gizmo es la que ofrece mayor varianza del tiempo entre paquetes. Los mayores valores los alcanza para las capturas realizadas durante la tarde y los menores que ofrece son durante la noche. Siendo en este último tramo donde varían más los valores de unas capturas a otras. Además esta aplicación presenta los mayores valores en tiempo medio entre paquetes. Con estos dos resultados podemos determinar que Gizmo no aporta buenos resultados en lo referente al tiempo entre paquetes.

La aplicación ooVoo va detrás de Gizmo, al igual que ocurre con el tiempo medio entre paquetes, pero en situaciones esporádicas le supera. Los mayores valores los obtiene durante la tarde pudiendo alcanzar incluso las 16 diezmilésimas, este máximo también lo es de todas las aplicaciones en todos los tramos horarios.

Los mínimos de varianza los ofrecen Skype, para los horarios de mañana y noche, ambos están en torno a las 0.5 diezmilésimas. La aplicación Damaka ofrece valores muy similares a los de Skype. Una característica de esta aplicación es que ofrece similares valores de varianza para cualquier tramo horario. VBuzzer cuando ofrece valores homogéneos para todas sus capturas es durante la tarde, siendo durante la mañana cuando obtiene sus mínimos.

Durante la noche es cuando se produce mayor variabilidad de unas capturas a otras. En general, podríamos decir que durante la mañana es cuando los valores que nos ofrecen las capturas son más homogéneos y de menor cuantía. Aunque los valores obtenidos son de unidades del orden de las diezmilésimas, nos muestran unas cantidades muy pequeñas de varianza. No obstante podríamos concluir que la mejor aplicación sería Skype tanto durante la mañana como por la tarde, ya que los valores ofrecidos tanto de varianza como de media del tiempo entre paquetes es mínimo. Las peores serían Gizmo y ooVoo, la primera por sus altos valores en todos los horarios y

la segunda, además de por sus altos resultados por lo que varían de una captura a otra. Ambos resultados se corresponden con el alto tiempo medio entre paquetes, lo que los hace no recomendables para su uso.

3.5.3 Ancho de banda medio.

En las siguientes gráficas podemos ver el ancho de banda en destino, para cada una de las cinco aplicaciones analizadas. Cabe mencionar que la aplicación Gizmo en las capturas realizadas durante la noche y la mañana no ha sido capaz de proporcionar video simultáneamente a la voz. Skype tampoco proporcionaba video durante las capturas de la mañana, debido a que resultaba imposible realizar la comunicación con video y voz. Además tenemos que recordar que la aplicación vBuzzer carece de video.

Capturas Mañana

Podemos observar (véase Fig.3.11) que los valores de todas las aplicaciones son aproximadas. Existen dos aplicaciones (ooVoo y Damaka) que sobresalen en alguna de sus capturas con respecto a los valores del resto. Estas son las mismas aplicaciones que más fluctúan de unas capturas a otras. La aplicación más homogénea en todas sus capturas es Skype.

- La aplicación Gizmo tiene sus valores compendidos entre 125 y 432 kbps. Se trata de una de las aplicaciones que ofrece los valores más bajos.
- La aplicación Skype vemos que es la más constante en todas sus capturas obteniendo un valor en todas ellas de 375 kbps aproximadamente.
- En la aplicación Damaka vemos que es la que llega a tener el mínimo, 50 kbps, pero es la más variable ya que sus valores fluctúan entre 50 y 1500 kbps. Obteniendo un valor distinto para cada una de las capturas.
- ooVoo vemos que es la aplicación ancho de banda requiere, alcanzando un valor máximo de 2050 kbps y un mínimo en torno a los 500 kbps.
- La aplicación vBuzzer utiliza un ancho de banda medio para recibir los paquetes de voz IP, que varían entre 500 y 650 kbps.

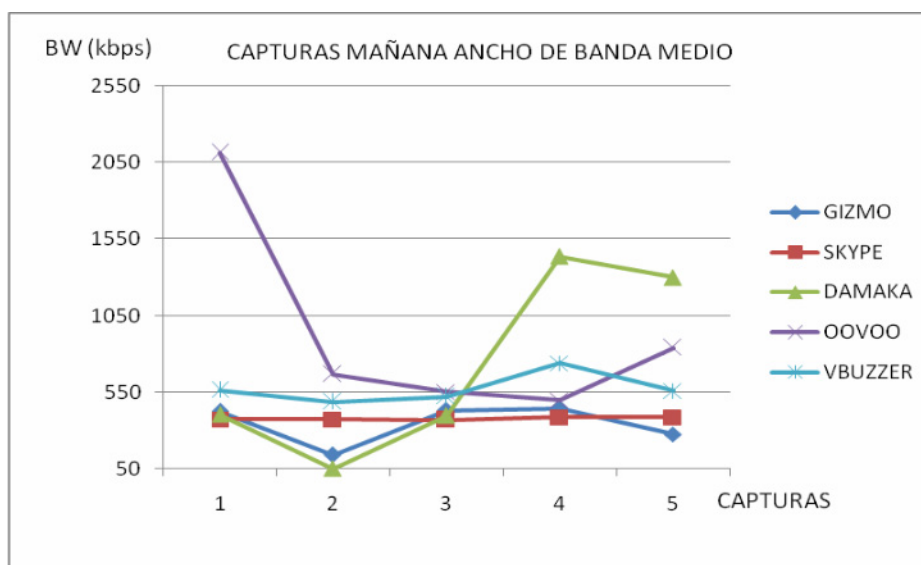


Figura 3.11 Gráfica del ancho de banda medio para las capturas realizadas durante la mañana.

Capturas Tarde

En las capturas realizadas durante la tarde (véase Fig.3.12) vemos que los anchos de banda en media son muy diferentes de unas aplicaciones a otras. Siendo la aplicación que más consume Skype, la más variable ooVoo, la más homogénea Damaka y la que menos ancho de banda requiere Gizmo.

- La aplicación Gizmo utiliza un ancho de banda comprendido entre 100 y 300 kbps. Es la que requiere menor ancho de banda en la recepción de paquete de voz IP. Sus valores son menores que los ofrecidos durante la mañana.
- La aplicación Skype es la que mayor ancho de banda utiliza, sobre 1700 kbps, sin embargo se mantiene constante este valor para las cinco capturas realizadas.
- La que proporciona mayor homogeneidad entre todas sus capturas es Damaka con un valor de 476 kbps.
- La aplicación ooVoo se caracteriza porque cada captura tiene valores muy diferentes a las anteriores. El margen de ancho de banda requerido está comprendido entre los 670 a los 1740 kbps.
- Finalmente, la aplicación vBuzzer se encuentra en el ecuador con un valor que oscila entre los 500 y los 700 kbps. Obtiene los mismos valores que los de la mañana.

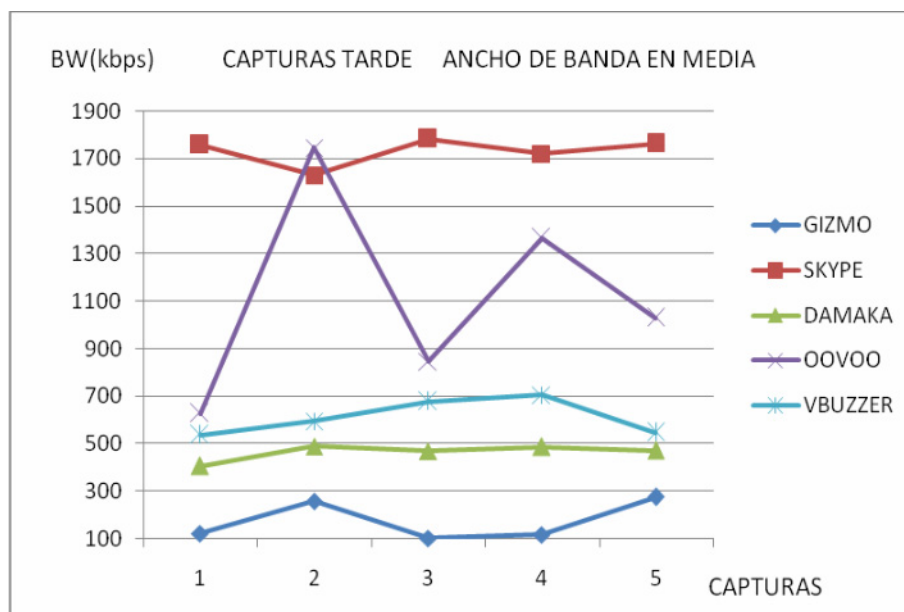


Figura 3.12 Gráfica del ancho de banda para las capturas realizadas durante la tarde

Capturas Noche

En las capturas realizadas durante la noche (véase Fig.3.13) podemos observar que la aplicación que utiliza menor ancho de banda para transmitir es Gizmo. Mientras que la que utiliza un mayor ancho de banda es Skype. También podemos observar que ooVoo es que proporciona mayor variabilidad en sus resultados.

- La aplicación Gizmo es la que menos ancho de banda consume, manteniéndose estos valores constantes para las cinco capturas realizadas con valores entre 7 y 12 kbps.
- La aplicación Skype hemos observado que es constante en todas sus capturas y presenta valores comprendidos entre los 3.5 Mbps y los 3.6 Mbps. Es la aplicación que proporciona mayor ancho de banda.
- La aplicación Damaka mantiene el ancho de banda utilizado constante para todas las capturas realizadas sobre un valor de 450 kbps.
- Con ooVoo también varían sus resultados de unas capturas a otras, aunque en menor medida que en Skype. El ancho de banda requerido en ooVoo va disminuyendo progresivamente, su máximo vale 2500 kbps y su mínimo 500 kbps. A excepción de Skype, esta aplicación es la que llega alcanzar mayores valores que el resto de aplicaciones.
- La aplicación vBuzzer varía ligeramente entre sus capturas, sus valores están comprendidos en el intervalo entre 360 y 773 kbps.

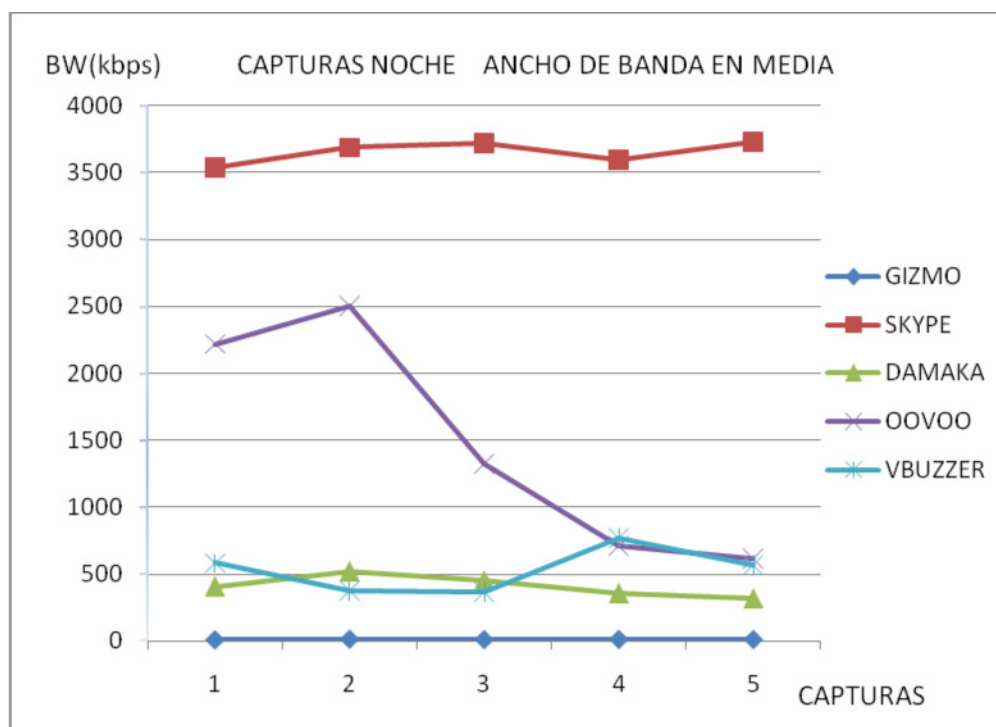


Figura 3.13 Gráfica del ancho de banda para las capturas realizadas durante la noche

Conclusión del ancho de banda medio

A la vista de los resultados podemos determinar que los mayores valores de ancho de banda que son necesarios para la transmisión de un paquete de voz se producen en la aplicación Skype en las capturas realizadas durante la noche. Y los menores también son durante la noche pero en la aplicación Gizmo. La mayor variabilidad de unas capturas a otras la alcanza ooVoo en los tres tramos horarios y la aplicación que toma iguales valores para cualquier horario es vBuzzer.

- La aplicación vBuzzer vemos que mantiene los mismos valores de ancho de banda en media para las tres franjas horarias.
- Observamos que Damaka cuando más varía es por la mañana, ya que durante la tarde y noche se mantiene constante para las cinco aplicaciones con los mismos valores en ambos tramos horarios.
- La aplicación Gizmo es más variable durante la mañana y también es en este horario cuando sus valores son máximos. Hemos observado que durante la noche es cuando Gizmo dedica el menor ancho de banda con un valor de 7 kbps. Esta aplicación es la que utiliza menos ancho de banda con respecto a las demás en cualquiera de los tres horarios.
- Sin embargo la aplicación Skype cuando adquiere los menores valores y se mantiene más constante es durante la mañana, influenciado porque sólo se pudieron realizar capturas de voz. Sin embargo durante la tarde y la mañana sigue manteniendo la homogeneidad entre sus capturas sin embargo aumenta el valor de ancho de banda requerido, siendo máximo durante la noche.
- La aplicación ooVoo no es constante para sus cinco capturas en ninguno de los tres horarios adquiriendo valores muy diferentes entre sí. Es imprevisible en ancho de banda que utiliza esta aplicación para la transmisión de un paquete de voz IP.

Si tuviéramos que elegir una aplicación según el ancho de banda que utiliza sería para la mañana ooVoo o vBuzzer ya que son las que ofrecen valores altos para todas sus capturas. Durante la tarde y la noche elegiríamos Skype, debido a que es la que utiliza mayor ancho de banda para transmitir los paquetes de voz IP.

3.5.4 Varianza del ancho de banda

Las gráficas realizadas de la varianza del ancho de banda hemos considerado que no tienen gran importancia por separado. Por lo tanto sólo comentaremos las conclusiones obtenidas al respecto.

Conclusión de la varianza del ancho de banda

La aplicación Gizmo ofrece menor varianza durante la noche, siendo en la mañana cuando obtiene sus máximos y cuando difieren más los resultados de unas capturas a otras.

La aplicación Skype vemos que durante la mañana posee valores mínimos. Estos resultados aumentan en las capturas de la tarde y de la noche, llegando a ser los máximos de entre las cinco aplicaciones. Aunque no existe gran diferencia en los resultados de unas capturas a otras.

Tanto Damaka, vBuzzer y ooVoo mantienen los mismos valores de su varianza en todos los tramos horarios, siendo en las dos primeras donde estos valores son constantes para todas las capturas y en ooVoo se producen diferencias entre unas y otras.

Podemos concluir que existe gran diferencia en las varianzas del ancho de banda obtenidos, para los tres tramos horarios. Los máximos al igual que los mínimos se obtienen para Skype y Gizmo, respectivamente, en el horario de la noche.

La aplicación que más varía para los tres tramos horarios es ooVoo y las más homogéneas son Damaka y vBuzzer.

3.5.5 Jitter medio.

En las siguientes gráficas podemos ver el *jitter* en media que se obtiene para cada una de las cinco aplicaciones en estudio.

Capturas Mañana

Para las capturas realizadas durante la mañana (véase Fig.3.14) podemos comprobar que su *jitter* obtiene valores distintos que podríamos separar en tres grupos. La aplicación Gizmo es la que más *jitter* tiene, la aplicación vBuzzer presenta poco *jitter* y las aplicaciones Damaka, ooVoo y Skype tienen un *jitter* prácticamente nulo.

- En la aplicación Gizmo el *jitter* tiene un valor de 8 ms, en la última de las capturas disminuye este valor hasta ofrecer 5 ms.
- La aplicación Skype ofrece un *jitter* prácticamente nulo. Al igual que las aplicaciones Damaka y ooVoo.
- En la aplicación vBuzzer los valores obtenidos se encuentran entre 0 y 1 ms.

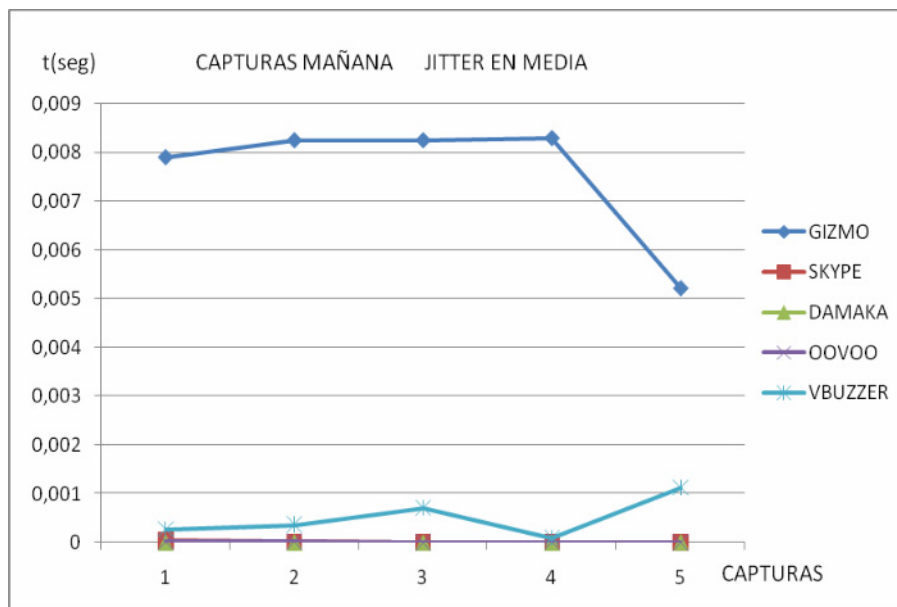


Figura 3.14 Gráfica del jitter medio para las capturas realizadas durante la mañana

Capturas Tarde

En las capturas realizadas durante la tarde (véase Fig.3.15) vemos que Gizmo es la única aplicación que tiene jitter, en el resto su valor es prácticamente nulo. Los valores que adquiere la aplicación Gizmo varían entre 30 ms y llegan a alcanzar 44 ms. Ha triplicado los valores obtenidos durante la mañana. En el resto de las

aplicaciones los valores obtenidos son del orden de los microsegundos, que hemos considerado prácticamente nulos.

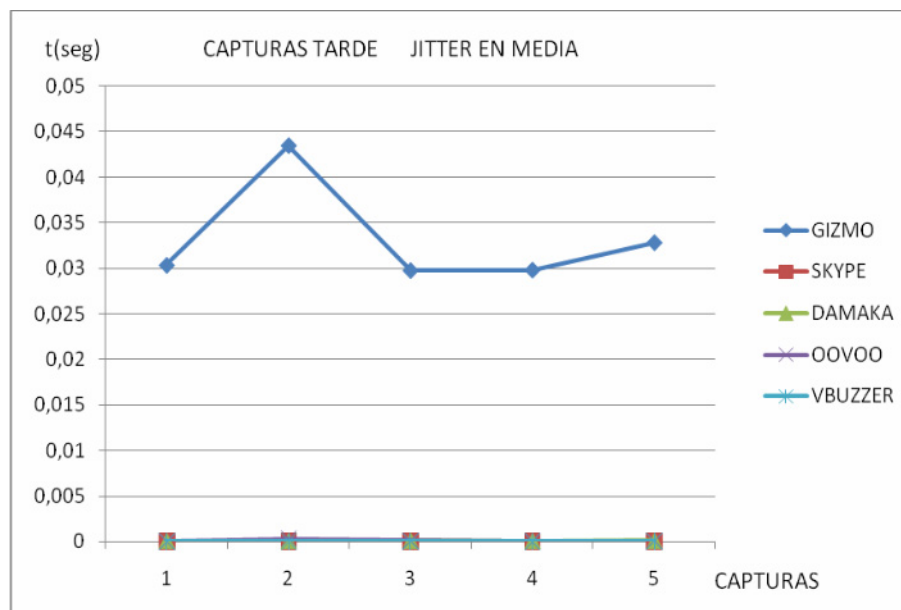


Figura 3.15 Gráfica del *jitter* medio para las capturas realizadas durante la tarde

Capturas Noche

En las capturas realizadas durante la noche (véase Fig.3.16) vemos que el *jitter* afecta sobre todo a las aplicaciones a Gizmo y a Skype, mientras que su valor en el resto de aplicaciones es prácticamente nulo.

- La aplicación Gizmo es la que posee los valores más altos de jitter, varían de los 4 ms a los 10 ms. Estos valores son ligeramente inferiores a los que ofrecía durante el horario matinal. Pero son considerablemente menores que los que ofrece durante la tarde.
- La aplicación Skype deja de tener los valores nulos que ofrecía en los otros tramos horarios. Estos valores fluctúan entre 0 y 6 ms.

Conclusiones del jitter

A partir de las gráficas anteriores, podemos concluir que la aplicación que ofrece *jitter* en todas las franjas horarias es Gizmo, siendo este valor mayor durante la tarde que aporta un *jitter* en media de 45 ms. Tanto por la mañana como por la noche el valor máximo que adquiere está alrededor de los 10 ms.

Existen otras aplicaciones que también tienen *jitter*, pero son menos importantes. Se trata de vBuzzer que presenta 1 ms de *jitter* en las capturas realizadas durante la mañana y de Skype durante la noche, con un valor aproximado de 5 ms.

Como conclusión obtenemos que la conversación no se ve afectada por el *jitter* en ninguna de las aplicaciones usadas ya que es prácticamente nulo en todos los tramos horarios, a excepción de Gizmo durante la tarde.

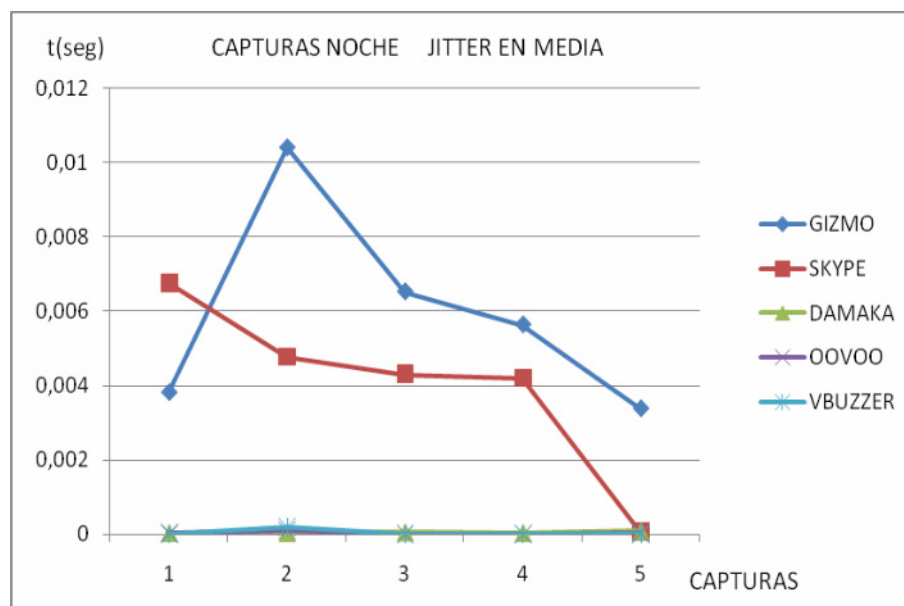


Figura 3.16 Gráfica del *jitter* medio para las capturas realizadas durante la noche

3.5.6 Varianza del *jitter*.

En las siguientes gráficas podemos ver la varianza del *jitter* que se obtiene para cada una de las cinco aplicaciones en estudio.

Capturas Mañana

En las capturas realizadas durante la mañana (véase Fig.3.17) observamos que las varianzas del *jitter* producidas en las cinco aplicaciones adquieren valores muy distintos y a la vez muy pequeños. Observamos que existen tres aplicaciones, Gizmo, ooVoo y vBuzzer, que tienen varianza de *jitter* máxima. La mínima la ofrece Skype y la aplicación que más fluctúa en los valores de sus cinco capturas es vBuzzer.

- El valor máximo de varianza del *jitter* lo ofrece Gizmo con 0,00036. Esta aplicación tiene su mínimo en 0,00025.
- La aplicación Skype tiene una varianza del *jitter* prácticamente nula para todas sus capturas.
- La aplicación Damaka es constante en tres de sus capturas con un valor de 0,0001, sin embargo para las otras dos capturas restantes su valor disminuye hasta 0,00005.
- La aplicación ooVoo tiene unos valores comprendidos entre 0,00023 y 0,00031. Es una de las aplicaciones que tienen una varianza del *jitter* máxima, para una de sus capturas.
- vBuzzer es la aplicación que más fluctúa, sus valores están comprendidos entre 0,00035, el máximo y 0,00008 el mínimo.

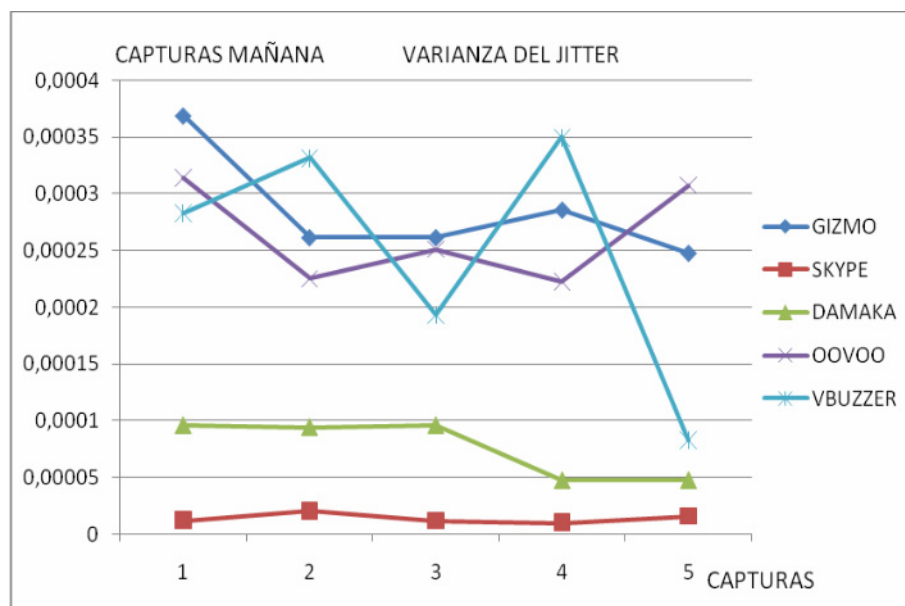


Figura 3.17 Gráfica de la varianza del *jitter* para las capturas realizadas durante la mañana.

Capturas Tarde

En las capturas realizadas durante la tarde (véase Fig.3.18) vemos que en el ooVoo la varianza del *jitter* adquiere valores distintos para cada una de las cinco capturas. Sin embargo, el resto de aplicaciones son más constantes para todas las capturas siendo Gizmo la que más valor tiene, el valor mínimo lo ofrece Skype y es cero, el resto las aplicaciones están en torno al 0,0001.

- La aplicación Gizmo tiene unos valores que varían del 0,001 al 0,0012. En cuatro de sus capturas es la que aporta el máximo valor de varianza. Esta aplicación adquiere valores mayores que los que ofrecía durante la mañana.
- Skype es la aplicación que ofrece el valor de varianza de *jitter* más bajo, es constante en todas sus capturas. Ofrece los mismos valores que los obtenidos durante la mañana.
- La aplicación Damaka tiene valores ligeramente superiores a los de Skype pero sigue la misma homogeneidad de ésta y los mismos valores que los obtenidos durante la mañana.
- ooVoo ofrece gran variabilidad entre los valores ofrecidos por unas y otras capturas. Esta aplicación nos da valores comprendidos entre 0,0003 y 0,0015, siendo este último el máximo valor de varianza de *jitter* que encontramos entre las cinco aplicaciones. El rango de valores que ofrece es mayor que el que ofrecía en las capturas realizadas durante la mañana.
- La aplicación vBuzzer, al igual que Damaka y Skype, se caracteriza por su homogeneidad en todas sus capturas con valores en torno a 0,00002. Podemos observar que tiene los mismos valores aproximadamente que los obtenidos durante las capturas matinales.

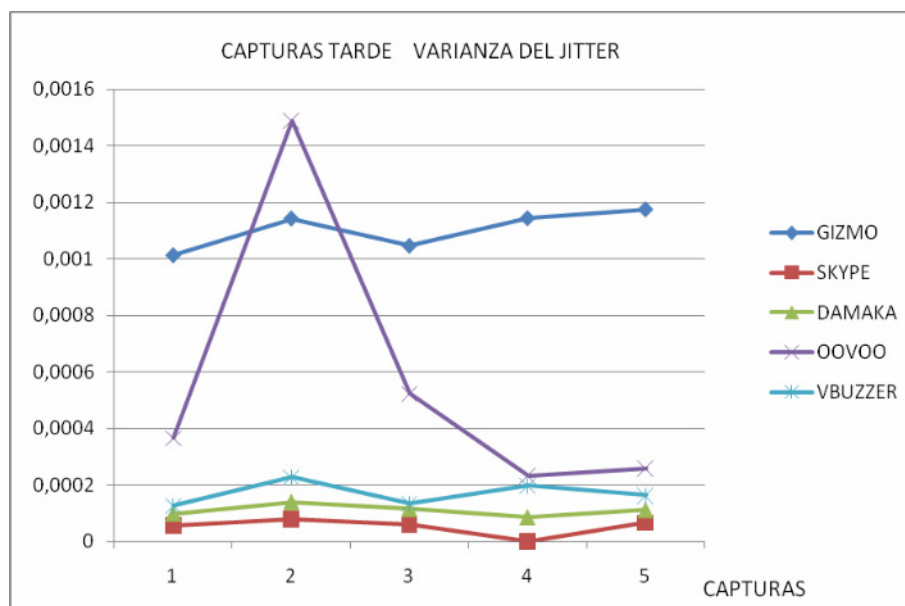


Figura 3.18 Gráfica de la varianza del jitter para las capturas realizadas durante la tarde

Capturas Noche

Vemos que en las capturas realizadas durante la noche (véase Fig.3.19) el *jitter* varía más que en el resto. Gizmo es el que mayor valor tiene. El resto de aplicaciones ya no está en torno al cero sino que aumentan su valor.

- La varianza del jitter en la aplicación Gizmo da valores que se encuentran entre 0,00008 y 0,00035. Estos valores son muy inferiores a los calculados para las capturas de la tarde, sin embargo son similares a los ofrecidos durante la mañana.
- La aplicación Skype incrementa su valor con respecto a los obtenidos durante la mañana y durante la tarde. Se encuentra dentro de los márgenes de 0,00005 y 0,00022. El mínimo se caracteriza por ser también el valor más bajo que se ha hallado para la varianza del *jitter* de las cinco aplicaciones durante la noche.
- La aplicación Damaka es la más homogénea con unos valores comprendidos entre 0,00009 y 0,00015. Estos resultados son similares a los obtenidos tanto durante la mañana como en la tarde.
- La aplicación ooVoo muestra unos valores inferiores a los obtenidos en las anteriores franjas horarias y con menor heterogeneidad entre sus capturas. Sus valores comprenden el margen entre 0,0001 y 0,00026.
- La aplicación vBuzzer es la que más varía en las capturas realizadas durante la noche, sus valores están comprendidos entre 0,00012 y 0,0004. Estos valores son similares a los obtenidos durante las capturas de la tarde y de la mañana.

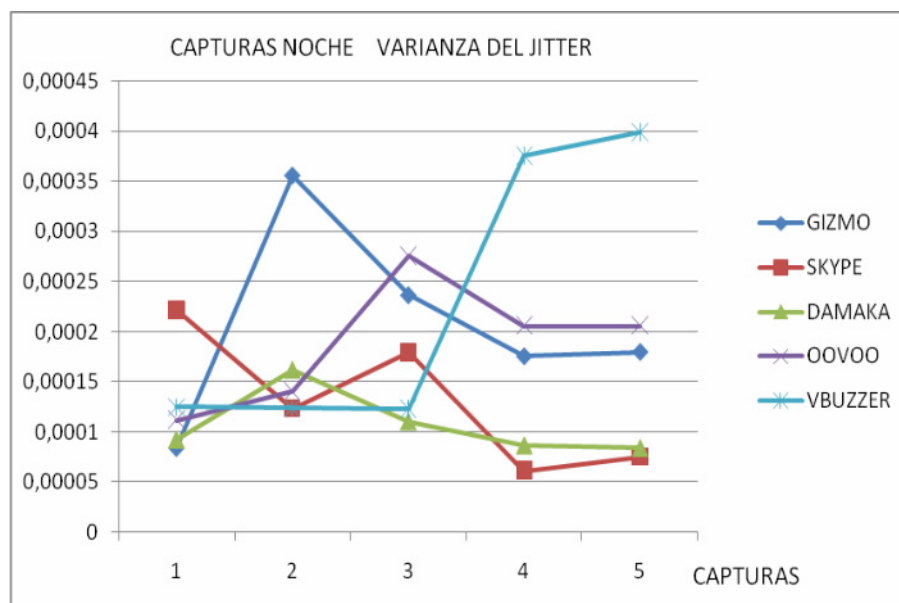


Figura 3.19 Gráfica de la varianza del jitter para las capturas realizadas durante la noche

Conclusiones de la varianza del jitter

En general podríamos decir que las aplicaciones que más se ven afectadas por la variación del jitter son Gizmo y ooVoo, las otras tres aplicaciones mantiene sus valores aproximadamente iguales para cada una de las tres franjas horarias. Skype es la que proporciona menor varianza del jitter para cualquier horario. Hemos observado que la aplicación que más fluctúa entre sus capturas es ooVoo en las realizadas durante la tarde. Y la más homogénea es Damaka para cualquier horario.

Podemos afirmar viendo los resultados anteriores que cuando se produce la mayor varianza de jitter es durante la tarde, siendo Gizmo la que proporciona mayores valores. También podemos decir que donde se produce la menor varianza del jitter es para las capturas realizadas durante la mañana.

3.5.7 Paquetes perdidos.

En las siguientes gráficas podemos ver el porcentaje de paquetes perdidos que se obtiene para cada una de las cinco aplicaciones en estudio.

Capturas Mañana

Durante la mañana (véase Fig.3.19) la aplicación que más paquetes pierde es Gizmo, siendo también ésta la más variable de todas. La que menos pérdidas ofrece es Skype. Las aplicaciones que no superan el 3% de paquetes perdidos son Damaka y Skype.

- Gizmo es la aplicación que más pérdida de paquetes ofrece, aunque sus valores son muy distintos de una captura a otra, varían de 3% al 10%.
- La aplicación Skype es la que menos paquetes pierde, no llega a superar el 1% y es la que durante la mañana se mantiene más constante.

- La aplicación Damaka también ofrece poca pérdida de paquetes, no supera el 2,5%. Sin embargo, es ligeramente más variable que Skype en cada captura realizada.
- La aplicación ooVoo ofrece unos valores bajos de pérdida de paquetes en torno al 1,5%, sin embargo en una de sus capturas llega a alcanzar el 4%. Es más variable que Skype y Damaka y llega a alcanzar mayores valores que éstas.
- La aplicación vBuzzer es constante en sus cinco capturas sin embargo los valores obtenidos están en torno al 3% y 4%, superiores a las tres aplicaciones anteriormente descritas. La ventaja que ofrece con la homogeneidad en todas sus capturas se ve disminuida por los porcentajes de pérdidas que obtiene.

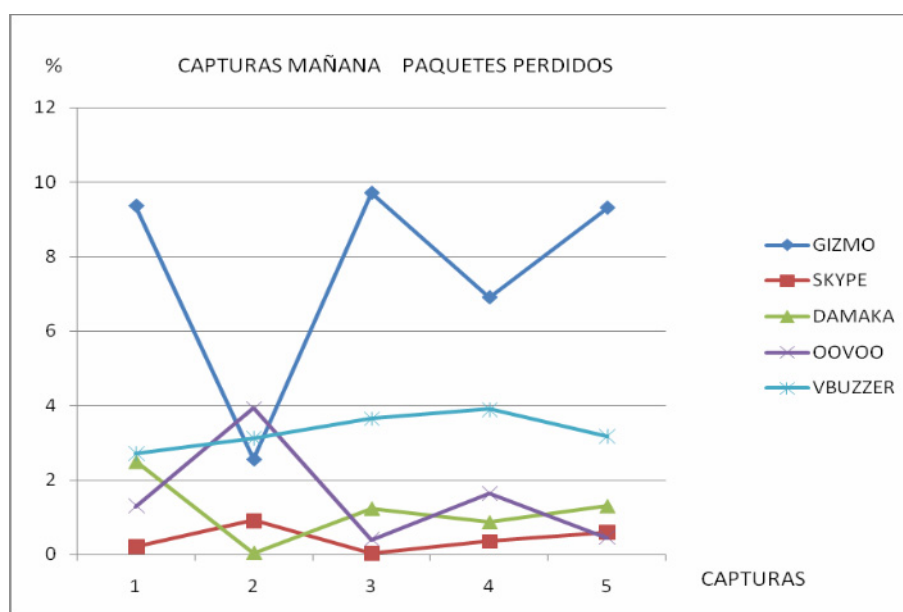


Figura 3.19 Gráfica del porcentaje de paquetes perdidos para las capturas realizadas durante la mañana

Capturas Tarde

En las capturas realizadas durante la tarde (véase Fig.3.20) la aplicación que más porcentaje de paquetes perdidos tiene es Gizmo. Para el resto de aplicaciones las pérdidas están sobre el 3%.

- La aplicación Gizmo es la que tiene mayor pérdida de paquetes. Cuatro de las cinco capturas ofrecen un valor en torno al 30%, sin embargo una quinta asciende hasta tener 58%.
- La aplicación Skype posee cuatro capturas cuyos valores no ascienden a más de 1% de paquetes perdidos, aunque existe otra captura en la que este valor se dispara y aumenta hasta un 15%.
- En Damaka todas las capturas se mantienen con el mismo valor, aproximadamente un 2%.

- En aplicación ooVoo tres de sus capturas ofrecen un porcentaje de 1% y las dos últimas aumentan este valor hasta el 2%. Ha disminuido tanto los valores como la heterogeneidad que ofrecía durante la mañana.
- La aplicación vBuzzer disminuye ligeramente los valores obtenidos durante la mañana, que bajan hasta alcanzar el 1%.

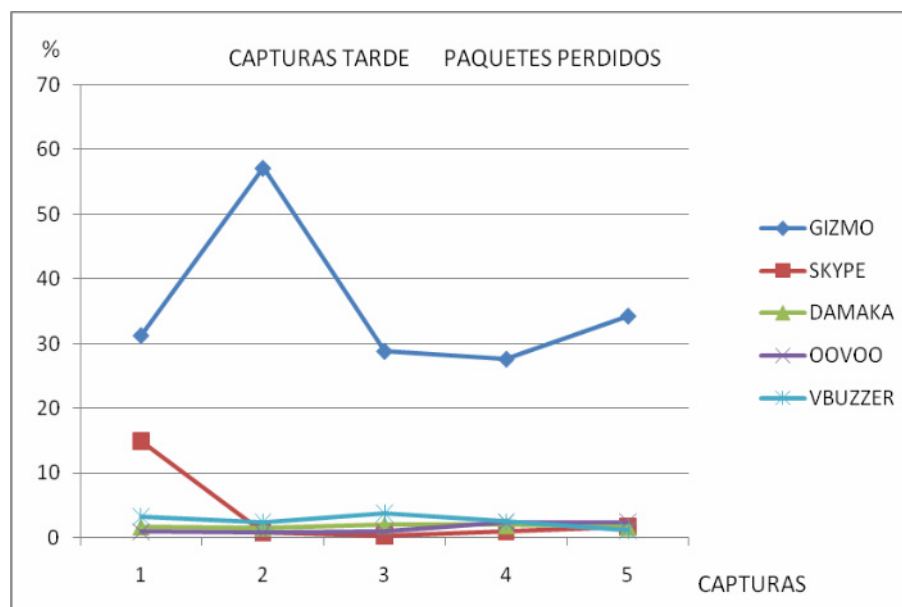


Figura 3.20 Gráfica del porcentaje de paquetes perdidos para las capturas realizadas durante la tarde.

Capturas Noche

En las capturas realizadas durante la noche (véase Fig.3.21) vemos que la aplicación que alcanza el mayor porcentaje de pérdidas es Gizmo y la que obtiene los menores valores es Skype.

- La aplicación Gizmo obtiene menores valores de pérdidas que en las capturas realizadas durante la tarde, pero mayores que las realizadas durante la mañana. Sus resultados adquieren valores comprendidos entre el 2% y el 21%. Vemos que mantiene la misma variabilidad que tenía durante la tarde.
- En la aplicación Skype el mayor porcentaje de paquetes perdidos es de 2,5%. Estos valores son mayores a los hallados en las capturas de la mañana.
- La aplicación Damaka es la que mantiene valores más similares entre las cinco capturas realizadas con un valor de 1,5%. Durante la noche obtiene valores más favorables que en el resto de día.
- En la aplicación ooVoo existen diferencias entre los valores de unas capturas y otras. El mínimo que alcanza es 0,01%, un valor muy bueno, sin embargo su máximo asciende hasta 8% (este valor deja de ser recomendable).

- La aplicación vBuzzer obtiene como valor máximo durante la noche un 3% y como mínimo un 0,5%. Obtiene valores más distintos entre sí que los encontrados en los otros horarios.

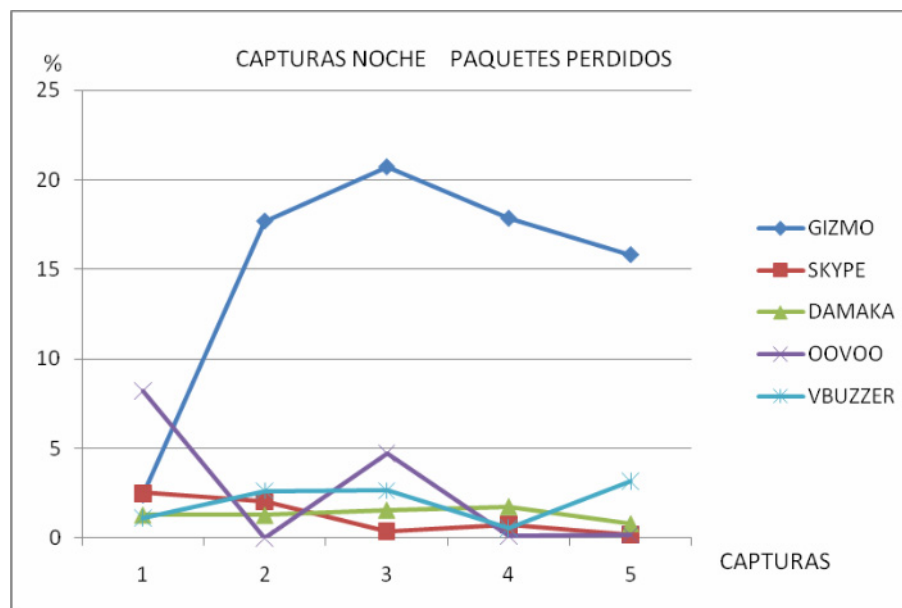


Figura 3.21 Gráfica del porcentaje de paquetes perdidos para las capturas realizadas durante la noche

Conclusiones de paquetes perdidos.

La aplicación Gizmo es la que más porcentaje de paquetes pierde en cualquiera de los horarios, por lo tanto podemos decir, que no es recomendable en ningún caso. La aplicación Skype es la que ofrece menores pérdidas durante la mañana, durante la tarde y la noche sus valores son mayores y más variados entre sí. Por su parte, la aplicación Damaka mantiene similares valores para todas las capturas realizadas, al igual que ocurre con vBuzzer. Mientras en ooVoo el porcentaje de pérdidas es mayor durante la noche y menor durante la tarde. Por la mañana y la noche es cuando adquiere valores más heterogéneos entre sus capturas.

Podemos concluir, teniendo en cuenta el porcentaje de paquetes perdidos, que durante la mañana elegiremos la aplicación Skype, para la tarde ooVoo y por la noche Damaka. La elección se realizaría buscando un porcentaje de paquetes perdidos mínimo o el más constante para las cinco capturas entre los mínimos ofrecidos.

3.6 Conclusiones

En general, teniendo en cuenta los 4 parámetros (tiempo entre paquetes, *jitter*, ancho de banda y pérdidas) las aplicaciones se podrían clasificar del siguiente modo:

- En primer lugar, Skype con un tiempo entre paquetes bajo, un jitter bajo, un ancho de banda alto y pérdidas casi nulas.

- En segundo lugar Damaka con un tiempo entre paquetes bajo, un jitter prácticamente nulo, un ancho de banda bajo y pérdidas aproximadamente nulas.
- En tercer lugar vBuzzer con un tiempo entre paquetes medio, un jitter aproximadamente nulo, un ancho de banda medio y pérdidas medias.
- En cuarto lugar ooVoo con un tiempo entre paquetes elevado, un jitter prácticamente nulo, un ancho de banda alto y pérdidas altas.
- En quinto lugar Gizmo con un tiempo entre paquetes muy elevado, un jitter alto, un ancho de banda muy bajo y pérdidas altas.

Capítulo 4

Análisis de QoE en aplicaciones de libre distribución

4.1 Introducción

Los creadores de aplicaciones tienen que tener en cuenta a la hora de crear una aplicación que el usuario final no es consciente de la tecnología con la que se implementa. Sin embargo sí es consciente de la calidad que percibe de la aplicación y cualquier valor inferior al que está acostumbrado le provocaría insatisfacción.

En nuestro caso las aplicaciones elegidas son Gizmo, Skype, Damaka, ooVoo y vBuzzer. Estas aplicaciones ofrecen servicios de voz y video, si la calidad de voz ofrecida es inferior a la que estamos acostumbrados con las PSTN, se produciría insatisfacción en este ámbito. Esto también ocurre con el video ya que la calidad ofrecida por los DVD es muy buena. Por tanto, si la aplicación nos ofrece resultados inferiores estarían provocando que un menor uso de la aplicación por parte de los usuarios.

Para realizar el estudio de la QoE que ofrecen las aplicaciones que estamos estudiando en este capítulo vamos a establecer la definición de QoE y todos los métodos con los que se puede medir. Seguidamente se realizará una descripción del entorno utilizado para la realización de las encuestas así como el material utilizado para ello. Mostraremos los resultados de las encuestas, que se hicieron a cien personas después de que éstas utilizaran cada una de las aplicaciones. Los resultados se presentan con una serie de gráficas que representan la puntuación obtenida para cada aplicación por parte de los encuestados.

Se incluye también un análisis de QoS a partir de las capturas realizadas simultáneamente con las encuestas y que emplearemos para comparar los resultados de QoE con los de QoS.

Finalmente vamos a establecer comparaciones entre los valores calculados para medir la calidad de servicio de la red y los calculados para evaluar la calidad que el usuario experimentaba de la aplicación. Para ello partiremos de los resultados de un trabajo realizado con anterioridad.

4.2 Definición QoE

Una definición de QoE sería cómo se perciben los resultados totales de un sistema desde el punto de vista del usuario. QoE mide la actuación final de todos los servicios desde la perspectiva del usuario. Es una medida que refleja cómo de bien la red satisface las necesidades del usuario final. No obstante, la QoE está relacionada con la QoS, la cual trata de realizar objetivamente mediciones del servicio entregado. Mientras que la QoE es una medida subjetiva de la experiencia que tiene el cliente de un servicio ofrecido.

El término de la calidad en la experiencia del usuario, QoE, comienza a ser importante a partir de los años 90 cuando el usuario, la experiencia que éste posee de la aplicación y el diseño centrado en sus necesidades comienzan a ser importantes para las empresas. Las empresas buscan la QoE debido a las ventajas que este factor les aporta. Si una aplicación proporciona su servicio con una alta QoE, los clientes estarán satisfechos, de lo contrario se le estará aportando al cliente una gran diferencia entre lo que él necesita y lo que la aplicación le ofrece. Si no se alcanza la QoE esperada por el usuario, se estará propiciando una pérdida de clientes ya que buscarán otras aplicaciones que le aporten la QoE que ellos esperan. Podemos determinar que hoy en día la calidad es una necesidad debido a la competición que existe entre aplicaciones.

Las medidas realizadas para determinar la QoE deben corresponderse directamente con la calidad que percibe el usuario de lo contrario la QoE hallada no se correspondería con la real. Existen numerosos factores que afectan a la calidad de la voz en la comunicación y como consecuencia a la QoE y a la QoS. A continuación los enumeraremos:

- Latencia: los paquetes de VoIP son muy sensibles a los retardos que se producen en la red.
- Jitter: Los paquetes de VoIP se encaminan al igual que el resto de paquetes por distintos nodos, sin embargo estos paquetes necesitan llegar en orden y que la diferencia de tiempo entre paquetes sea similar a la diferencia con la que se transmitieron.
- Pérdida de paquetes: la voz se transmite en paquetes, si estos paquetes se pierden, también se pierde el trozo de conversación que se transportaba en ellos.
- Inestabilidad de las llamadas: se produciría cuando una vez establecida una llamada, se produce inesperadamente una finalización de la conexión. Se debe al fallo de algún equipo o a la congestión de la red.
- Bloqueo de las llamadas: también se debe a la congestión de la red y se produce cuando se quiere iniciar una llamada pero esta no se consigue conectar.

Los factores que producen impacto en la QoE son:

- Factores Humanos: la expectación que tienen de la aplicación, la edad del usuario, la familiaridad que tiene con las aplicaciones,...
- Compresión y codificación de la voz: el algoritmo usado para comprimir la voz, el algoritmo para codificar la voz, la supresión del silencio, la cancelación del eco...
- Temas de la red: retraso, variación de retraso, paquetes perdidos, disponibilidad de la red
- Características de servicio: ancho de banda disponible para transmitir la llamada, características de las llamadas, número de llamadas, *Wireless* o conexión de cable.

4.3 Medir la QoE

La QoE es normalmente medida en términos de medidas técnicas, ignorando el hecho que la última meta no debería ser desarrollar aplicaciones con la más

avanzadas características, sino desarrollar productos que aseguren una buena calidad de la experiencia del usuario final.

La calidad de la experiencia de los humanos cambia con el tiempo porque diferentes factores contextuales influyen en ello. Como consecuencia, la QoE debería ser medida en toda su dimensión.

Un dato a tener en cuenta es que en aplicaciones en las que se producen videoconferencias tiene que existir una sincronización entre los componentes de audio y video para que puedan asegurar una satisfactoria QoE.

4.3.1 Medir la QoE de video

Las contribuciones de la calidad de las imágenes de video a QoE pueden ser medidas de tres maneras:

- Subjetivamente: usando un experimento controlado y participantes que gradúen la calidad usando escalas como MOS.
- Objetivamente: usando equipos electrónicos para medir varios aspectos de la calidad de la señal de video
- Indirectamente: usando medidas de los deterioros de la red (pérdida, retrasos, jitter, duración del defecto) para estimar el impacto en la calidad de video, donde existe una relación establecida entre QoE y QoS.

Medida subjetiva de la calidad de video

El encargado de medir la calidad de la imagen de video es la persona que lo ve. La meta de cualquier servicio de video es complacer al usuario con imágenes de alta calidad y servicio. Los usuarios cada vez más están empezando a realizar juicios sofisticados en la calidad de video con el servicio como referencia de DVD. Este método proporciona las valoraciones más exactas de calidad de video para la perspectiva del servicio proporcionado a los clientes.

En general, las valoraciones del usuario dependen de muchos factores: distancia, tamaño y resolución del video, brillo, contraste, saturación del color, distorsión, naturalidad y definición. Los usuarios son particularmente sensibles a la compresión digital y a las pérdidas introducidas en las redes digitales que dan como resultado distorsión en el video y ofrecen secuencias de imágenes antinaturales.

Las encuestas subjetivas son con frecuencia una media de la puntuación de calidad llamada MOS.

Medida objetiva de la calidad de video

Estas técnicas, aunque no son tan exactas como las subjetivas, ofrecen un buen compromiso en las valoraciones de la calidad de video. Podemos establecer resultados como combinación de ambas ya que las mediciones objetivas y subjetivas son complementarias.

Las técnicas empleadas son clasificadas en cuatro categorías:

- Técnicas basadas en modelos de percepción humana de video
- Técnicas basadas en parámetros en la señal de video: compara la señal de video recibida con la transmitida
- Técnicas basadas en los parámetros deteriorados de la red

- Técnicas basadas en la duración de los deterioros en la señal de video

Medida indirecta de la calidad de video

Este método usa los parámetros de los paquetes de la red como son el tiempo de llegada de paquetes, retraso, jitter, pérdida y la duración de los fallos. Para ello monitoriza las transmisiones realizadas en la red y se calculan los anteriores parámetros.

4.3.2 Medir la QoE de la voz

Existen diversos métodos que se utilizan para medir la calidad de la voz. La mayoría de ellos lo que intentar es aproximar la calidad de servicio que ofrece la aplicación a la calidad que experimenta el usuario aplicándole a los datos hallados un factor de conversión. Esto se debe a que realizar la QoE directamente con los usuarios es una tarea difícil.

Cada uno de los factores anteriores que contribuyen a la QoE puede medirse individualmente. Sin embargo, resulta útil tener un indicador que refleje la totalidad de la calidad de voz que se ofrece.

Existen tres formas más comunes de medir la QoE y son:

- La medida subjetiva llamada MOS.
- Un estimador objetivo de MOS llamado PESQ (*Perceptual Evaluation of Speech Quality*, Evaluación de la Calidad Vocal por Percepción) [ITU-T P.862].
- Medida computacional llamada *Transmission Rating* (R, Índice de Transmisión) [ITU G.107], es calculada a partir de las medidas objetivas de quince parámetros usando la herramienta del estándar ITU G.107 llamada Modelo E.

MOS

Es directamente medido de la percepción por parte del usuario de la calidad de la voz, es una medida directa de la QoE. Puede ser obtenida mediante la realización de encuestas a los usuarios. Las puntuaciones obtenidas de la calidad son evaluadas sobre cinco puntos:

- Excelente (5)
- Bueno (4)
- Pasable (3)
- Pobre (2)
- Malo (1)

MOS es calculado por la media de todas las puntuaciones dadas para cada caso. El mayor valor para MOS (5) refleja la mejor percepción que se ha obtenido de la calidad.

Los inconvenientes de determinar la QoE mediante el método MOS son que se ve afectado por el contexto del experimento y que la realización de estudios subjetivos requiere una cantidad considerable de tiempo y esfuerzo.

PESQ

Los estimadores de MOS como es PESQ pueden proporcionar una rápida y repetida estimación de la distorsión en la señal. Sin embargo, la puntuación no refleja la calidad de la voz durante la conversación, ya que el retraso y el eco son excluidos del cálculo.

P.862 es un método encargado de realizar el estudio de un canal. Compara una señal inicial $X(t)$ con una señal degradada $Y(t)$ que se obtiene como resultado de la transmisión de $X(t)$ a través de un sistema de comunicaciones. La salida de PESQ es una predicción de la calidad percibida por los sujetos en una prueba de escucha subjetiva y que sería atribuida a $Y(t)$.

Para las dos ondas que tengan mayor similitud y que obtengan menor distorsión se les asignará la mejor puntuación.

La gama de la nota PESQ es de 0,5 a 4,5, si bien en la mayoría de los casos la gama de salida será la de una nota similar a la MOS de calidad de escucha entre 1,0 y 4,5, que es la gama normal de los valores de las notas MOS.

TRANSMISSION RATING (R)

Es una medida objetiva que indica toda la calidad de voz. R es la principal variable de salida del modelo E de ITU. Para calcular R son necesarios quince parámetros, algunos de los cuales son el nivel de escucha, ruido, distorsión, los codificadores usados, paquetes perdidos, retraso y eco. R tiene en cuenta todos los factores que contribuyen a la calidad de la voz en una conversación, es el único valor necesario para describir completamente la calidad.

Los valores usados para calcular R pueden ser valores medidos o valores esperados. Esto significa que el modelo E puede ser utilizado para predecir la calidad de los equipos y las redes que todavía no se han realizado. Es también útil calcular R mediante la referencia de una red. Unas referencias útiles para ello serían:

- El TDM de una PSTN
- Una red existente que está siendo reemplazada por una nueva red.
- Una red que desarrolla una QoE conocida puede ser elegida para servir como una tarjeta de calidad.

El modelo E se basa en el método de factor de degradación del equipo, de acuerdo con modelos de determinación de índices de transmisión previos. En una primera etapa, el resultado de cualquier cálculo con el modelo E es un factor de determinación de índices de transmisión R , que combina todos los parámetros de transmisión pertinentes para la conexión considerada. Este factor R está constituido por:

$$R = R_0 - I_s - I_d - I_{e-eff} + A$$

R_0 representa en principio la relación señal/ruido básica que incluye fuentes de ruido, tales como ruido de circuito y ruido ambiente. El factor I_s es una combinación de todas las degradaciones que aparecen de forma más o menos simultánea con la señal vocal. El factor I_d representa las degradaciones producidas por el retardo y el factor de degradación efectiva del equipo I_{e-eff} representa las degradaciones producidas por codificadores de velocidad binaria baja. Incluye también la degradación debida a pérdidas de paquetes de distribución aleatoria. El factor de mejora A permite compensar los factores de degradación cuando existen otras ventajas de acceso para

el usuario. El término *Ro* y los valores *Is* e *Id* se subdividen en valores de degradación específicos más detallados.

4.4 Entorno de las mediciones

A continuación vamos a describir las principales características que componían en entorno en el que se realizaron las encuestas. Esto nos ofrecerá una visión de lo que los dispositivos de voz y video han podido afectar a la calidad de la llamada o si por el contrario no le han aportado ningún inconveniente.

4.4.1 Vídeo

Vamos a comparar el entorno en el que se realizaron las encuestas con el que establece la ITU P910 como óptimo, para medir la calidad de video en este tipo de conexiones.

Luz

La ITU-T establece que las fuentes de luz, bombillas o tubos fluorescentes, pueden ser emplazadas arriba o al lado de la cámara. La variación de la luz debe ser mínima porque puede causar parpadeos en la secuencia de video, afectando de este modo a su calidad. Otros factores importantes son los colores de la pared, superficie reflejante,... deben ser cuidadosamente controladas.

Las encuestas se realizaron en la biblioteca de la Universidad Politécnica de Cartagena (UPCT) donde la luz es ofrecida por tubos fluorescentes colocados en el techo, poseen muy poca variación de luz. Esto es una ventaja ya que este factor afecta poco a la calidad obtenida de la imagen.

En el fondo de la imagen se mostraba la pared, estas están formadas por piedras marrones. Podemos concluir entonces que no existía superficie reflejante que pudiera afectar a la imagen ofrecida.

Cámara

La ITU-T recomienda que las secuencias de imágenes sean grabadas por una cámara de alta calidad. Se pueden usar sistemas de longitud focal fijos o variables. Para los terminales de escritorio una profundidad focal entre 30 y 120 cm es razonable, mientras que para sistemas multiusuario la profundidad debe ser como mínimo de 50 cm. Para soportar la variación de la iluminación en la habitación deben ser usados un iris ajustable o filtros de densidad neutral. La cámara debe tener un balance automático de blancos para pueda acoplar el color de la fuente de luz.

Las cámaras utilizadas para realizar la imagen de video tienen el inconveniente de no ser de alta calidad como recomienda la ITU. Se trataba de cámaras de 1,3 mega píxeles. No obstante tienen la ventaja de que pueden soportar la variación de la iluminación y que posee un balance automático de blancos.

La distancia a la que se situaban los encuestados de la cámara se encontraba dentro del margen establecido, ya que se situaban entre 30 y 40 cm de ella.

ACR

El método ACR (*Absolute Category Rating*, Categoría de Índice Absoluto) especifica que para cada presentación visual el encuestado evaluará la calidad de la imagen mostrada siguiendo una escala de cinco niveles:

- Excelente

- Bueno
- Aceptable
- Pobre
- Mala

Esta forma de evaluar la calidad de la imagen fue la utilizada por los encuestados para puntuar las imágenes ofrecidas por cada una de las aplicaciones.

Pantalla

Tanto el tamaño y tipo de pantalla usados deben ser apropiados para la aplicación que se esté estudiando. Cuando las secuencias de imagen son presentadas a través de un PC, las características deben ser especificadas. Por ejemplo, el tipo de monitor usado o tarjeta de video usada. Es preferible usar la misma pantalla para ver las imágenes. También es importante especificar que el color de fondo de la pantalla debe corresponder con un 50% de gris.

Las pantallas utilizadas para nuestro estudio fueron unos monitores de PC con una calidad del color de 32 bit, una resolución de la pantalla 1280x800 y una frecuencia de actualización de 60 Hertz. La tarjeta de video que utiliza es el Mobile Intel(R) 945GM Express Chipset Family.

Sin embargo, las aplicaciones mostraban la imagen de video en un tamaño menor al de la pantalla, de 320x240.

La distancia a la que los encuestados se encontraban de la pantalla era de aproximadamente unos 30 cm.

Encuestados

El número de encuestados necesarios para realizar encuestas sobre imágenes de video es de 4 a 40. El número de encuestados depende realmente de la validación necesaria y la necesidad de generalizar una aplicación a una gran población. Las características que tienen que poseer estas personas es que no tienen que trabajar en temas relacionados con los de calidad de video y no se le debería ofrecer asesoramiento por parte de los que si tienen experiencia en este sector.

En nuestro caso, se cumplen las características que tienen que poseer los encuestados ya que ninguno de ellos tenía experiencia o había trabajado en temas relacionados con la calidad de video y no recibieron asesoramiento por nadie.

También se cumple el número de personas a las que se les pidió su opinión ya que fueron 100 encuestados, todos estudiantes de universidad.

4.4.2 Voz

Para establecer las condiciones en las que se medía la voz vamos a estudiar cómo se transmitió y los dispositivos de interfaz humana usados para ello.

Micrófono.

El micrófono utilizado poseía gran sensibilidad a los sonidos, además como su colocación era muy cercana a la boca, unos 5 cm, la captura de la voz era total.

Esto nos hace pensar que todos los paquetes perdidos no son causa de la mala calidad del micrófono utilizado sino de la aplicación usada.

Altavoces.

Para poder escuchar el sonido se utilizaron dos auriculares para evitar la interferencia de sonidos en el oído del encuestado.

Así los ruidos ambientales, aunque eran pocos, no se mezclaban con la comunicación de VoIP establecida mediante cada una de las aplicaciones.

Ruido del entorno.

Las encuestas se realizaron dentro de la biblioteca de la UPCT, no existían ruidos en el entorno que pudieran interferir en la señal ya que predominaba el silencio.

De este modo podemos afirmar que los ruidos que el usuario percibe durante las encuestas son introducidos por la aplicación.

4.4.3 Otros datos de interés

Las llamadas se realizaron entre dos dispositivos pertenecientes a la misma red local, más concretamente estaban dentro de la red local de la biblioteca de Antigonos de la UPCT.

Los ordenadores se encontraban dentro de habitáculos distintos para que fuera imposible escuchar a la otra persona por otro medio que no fuera mediante la aplicación de VoIP utilizada. También hay que mencionar que la conexión a red era *Wireless*.

Los dispositivos utilizados para escuchar eran auriculares de esta manera se garantizaba que el sonido se transmitía directamente al oído de interlocutor, impidiendo que se pudieran producir errores debido a los altavoces de mala calidad de los ordenadores.

La duración de las llamadas realizadas fue de aproximadamente un minuto y medio, tiempo suficiente para hacer una valoración individual de la aplicación y comprobar sus características.

Las encuestas fueron realizadas por gente joven cuyas edades estaban comprendidas entre los 19 y 28 años. Todos los encuestados eran estudiantes universitarios que estaban estudiando para los exámenes de junio. Aproximadamente el 95% de los encuestados sólo conocía las aplicaciones Skype y Gizmo aunque sólo un 7% de ellos las había utilizado alguna vez. El 5% restante sólo conocía Skype aunque no había usado esta aplicación nunca.

4.5 Análisis de QoS de las conversaciones de los encuestados

Simultáneamente que se establecía comunicación con el otro extremo mediante la aplicación de VoIP se utilizaba la herramienta *Wireshark* para realizar capturas de cada una de las llamadas realizadas. Al igual que hicimos en el capítulo anterior, estas capturas fueron pasadas a un formato simple y mediante la herramienta *awk* se obtuvieron los resultados de *jitter*, ancho de banda, tiempo entre paquetes y paquetes perdidos.

Sin entrar en detalle mostraremos los valores obtenidos para cada una de las aplicaciones.

Gizmo

- Las capturas realizadas en Gizmo muestran que el tiempo medio entre paquetes resultó que en todas las realizadas tienen un valor como mínimo superior a 70 ms.
- El ancho de banda medio obtiene valores muy variables que oscilan entre 60 kbps y 10 Mbps.

- El valor de jitter medio obtenido para el 70% de los casos no llega a superar los 5 ms. El 30% restante son valores muy variables que componen el rango entre 30 ms a 180 ms.
- El porcentaje de paquetes perdidos obtiene valores muy altos ya que sólo el 24% de las capturas ofrecen un valor inferior a 3%, un 57% obtiene valores comprendidos entre 3% y 15% y un 19% obtiene valores superiores a 50% de paquetes perdidos.

Skype

- Para la aplicación Skype, en el cálculo del tiempo medio entre paquetes se obtiene muy buenos resultados ya que para el 98% de los resultados es inferior a 20 ms.
- El ancho de banda medio da resultados que varían entre 700 kbps y 3,8 Mbps.
- En los resultados de jitter medio también encontramos valores muy pequeños ya que el 80% de las encuestas realizadas muestran valores inferiores a 1 ms. En el otro 20% los resultados están entre 1 ms y 5 ms.
- Para el 75% de las capturas el porcentaje de paquetes perdidos es inferior a 3% y un 25% corresponde a valores superiores a 3% que llegan a alcanzar incluso el 20% de pérdidas.

Damaka

- El tiempo medio entre paquetes resultante en las capturas de Damaka se divide en un 87% que posee valores inferiores a 20 ms y un 13% que llega a alcanzar incluso los 100 ms.
- El ancho de banda medio utilizado para transmitir los paquetes de VoIP varía entre 200 kbps y 500 kbps.
- Para esta aplicación se obtiene un jitter medio donde el 96% de las aplicaciones no llegan a alcanzar 1 ms, sólo un 4% de las capturas de las encuestas ofrecen un jitter medio superior a 1 ms alcanzando como valor máximo 6 ms.
- El porcentaje de paquetes perdidos calculado obtiene valores inferiores a 3% para el 76% de las capturas realizadas. El 14% restante obtiene valores superiores a 3% y que llegan a alcanzar como máximo un 30% de pérdidas.

ooVoo

- Para ooVoo el tiempo medio existente entre paquetes está en torno a los 55 ms para todas las capturas.
- Las capturas realizadas para esta aplicación obtiene valores muy diferentes entre sí para el ancho de banda medio ya que establece un rango de valores comprendido entre 100 kbps y 7 Mbps.
- Para el 90% de las aplicaciones el jitter medio obtiene valores inferiores a 1 ms, el otro 10% adquiere resultados que varían entre 5 ms y 90 ms.
- El porcentaje de paquetes perdidos que ofrece esta aplicación es menor de 3% para el 78% de las capturas, para el 17% se obtiene valores

comprendidos entre 3% y 10%, el restante 5% las pérdidas son superiores al 50% de los paquetes transmitidos.

vBuzzer

- El tiempo medio entre paquetes resultante para las capturas realizadas con vBuzzer vale aproximadamente 26 ms para el 87% de las capturas. En el 13% restante se obtienen valores comprendidos entre 40 ms y 90 ms.
- El ancho de banda que nos muestran las capturas realizadas con vBuzzer obtiene valores muy distintos que varían entre 400 kbps y 1 Mbps.
- El jitter medio obtenido para vBuzzer es inferior a 1 ms para el 89% de las capturas realizadas, el 11% restante obtiene valores superiores a 1 ms y como máximo llegan a valer 4 ms.
- Se obtiene un porcentaje de paquetes perdidos inferior al 3 % para el 88% de las capturas de las encuestas, el 12% varía entre pérdidas de 4% y llegan a alcanzar valores de 28%.

Vamos a realizar una pequeña comparación entre los valores obtenidos al medir la QoS en el capítulo 3 con los resultados anteriores correspondientes a las encuestas. Para el tiempo medio entre paquetes los valores son similares a los obtenidos en las capturas correspondientes al estudio de la QoS. Observamos que las aplicaciones Gizmo, ooVoo y vBuzzer tienen un valor de ancho de banda muy superior a los resultados medidos para la QoS. Sin embargo, Skype y Damaka se mantienen en los mismos valores que en el estudio anterior. Otras características importantes es que el jitter medio y el porcentaje de paquetes perdidos obtienen valores muy superiores a los alcanzados mediante el estudio de la QoS.

4.6 Resultados

Para cada pregunta realizada en la encuesta se muestran los diferentes resultados obtenidos para cada aplicación.

4.6.1 Resultados que espera el usuario de la aplicación.

Los resultados mostrados a continuación (Fig.4.1) representan el nivel esperado de servicio que el usuario cree que le puede ofrecer cada aplicación. Se puede apreciar que en las aplicaciones más conocidas como son Skype y Gizmo se espera un nivel máximo de calidad en la aplicación. Aunque es menos conocida de ooVoo también el usuario espera un gran servicio.

En la aplicación Skype el 73% de los encuestados esperan un servicio máximo, sin embargo en vBuzzer se obtiene que sólo el 7% creen que pueden recibir un servicio excelente de esta aplicación.

Existe una gran expectación de los usuarios para las aplicaciones de VoIP ya que en todos los casos se espera un nivel bueno de servicio superior al 70%.

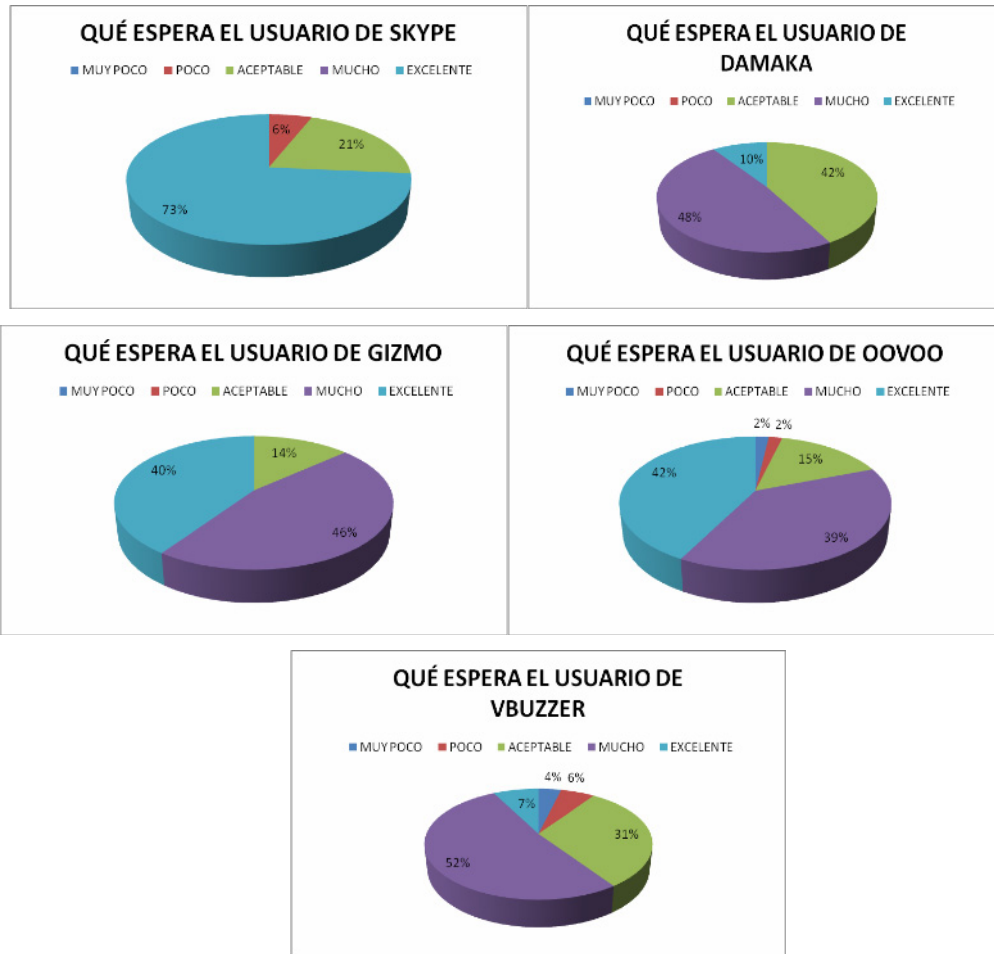


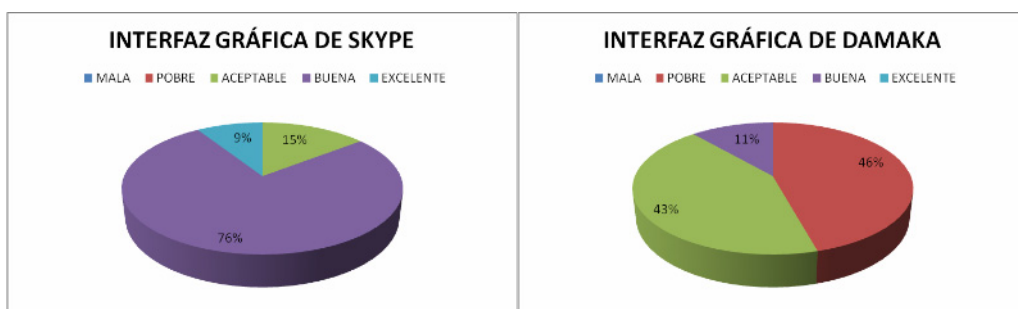
Figura 4.1 Resultados que espera el usuario en cada una de las aplicaciones.

4.6.2 Interfaz gráfica

La valoración que el usuario ha hecho sobre la interfaz gráfica de la aplicación (Fig.4.2) es considerada excelente con un 70% para la aplicación ooVoo. Sin embargo, Damaka es la que obtiene peores resultados ya que el 46% de los encuestados consideran que su diseño es muy pobre.

La segunda aplicación que tiene mejor valorada su interfaz gráfica es Skype donde el 76% de los encuestados considera que es buena.

En las aplicaciones Gizmo y vBuzzer los resultados varían entre los que consideran que son aplicaciones con una pobre interfaz gráfica, aceptable y buena. La más destacadas son las calificaciones como aceptables con un 62% y 64% respectivamente.



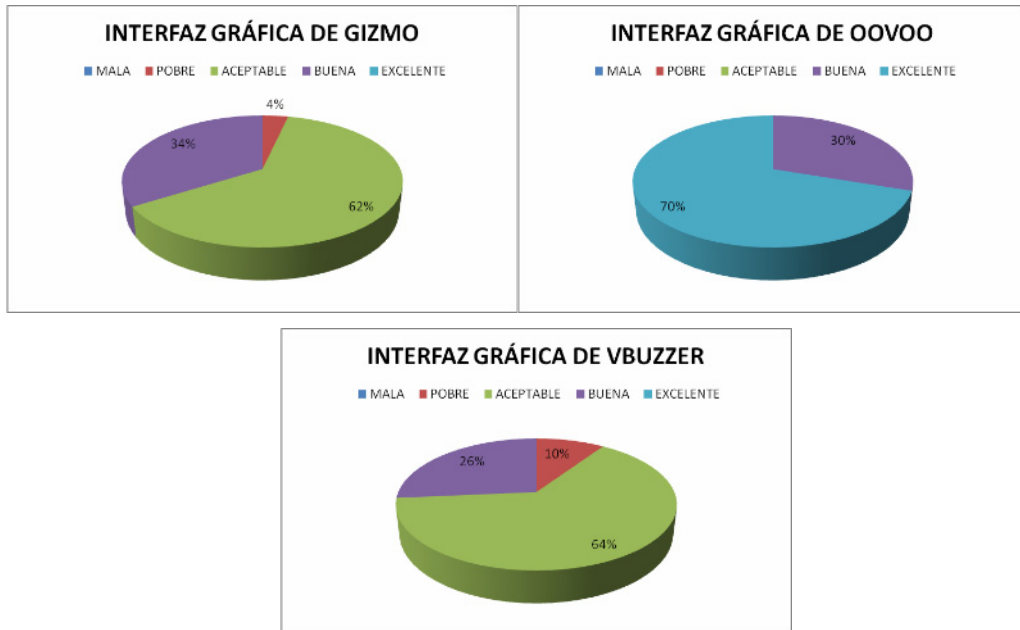


Figura 4.2 Valoración del usuario de la interfaz gráfica de cada aplicación.

4.6.3 Tiempo que tarda la aplicación en conectarse.

Podemos apreciar (Fig.4.3) que la aplicación que consideran los encuestados que tarda en conectarse un tiempo excesivo es Damaka con un 48%. Por otro parte, también encontramos que las aplicaciones que tardan muy poco, según los encuestados, son ooVoo y vBuzzer con un 94% y 88% respectivamente.

Las aplicaciones Skype y Gizmo mayoritariamente son consideradas como aplicaciones que tardan poco o muy poco en conectarse con un 78% y 86%, respectivamente.

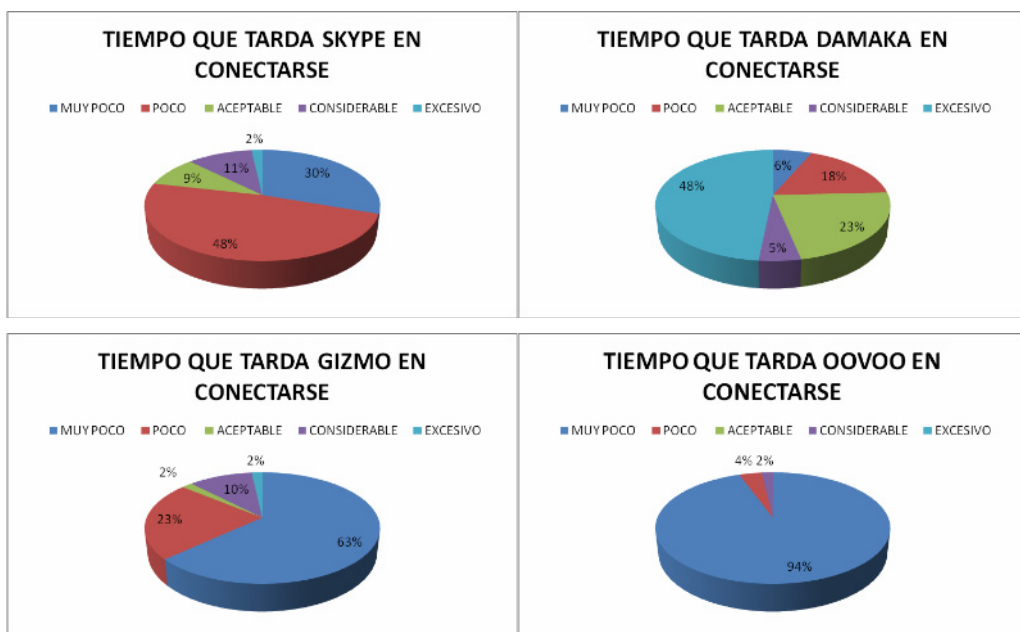




Figura 4.3 Valoración del usuario del tiempo que tarda en conectarse cada aplicación.

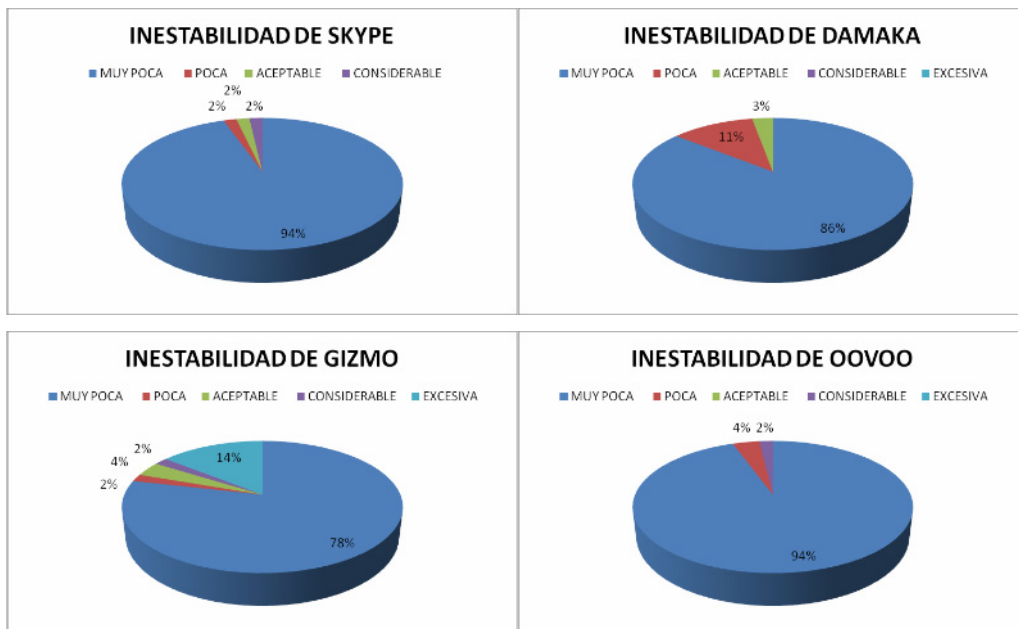
4.6.4 Inestabilidad.

En las siguientes gráficas (Fig.4.4) observamos que todas las aplicaciones son principalmente muy estables.

Los mayores valores de estabilidad son alcanzados por Skype y ooVoo con un 94% y les sigue vBuzzer con un 93%.

Los peores valores los encontramos en Gizmo donde el 14% de los encuestados considera que la inestabilidad producida en la aplicación es excesiva, sin embargo este valor es pequeño en comparación con el 78% de los usuarios que consideran que su inestabilidad es mínima.

Para la aplicación Damaka los valores obtenidos son muy buenos ya que el 97% de los encuestados considera que la inestabilidad que existe en este programa es poca o muy poca.



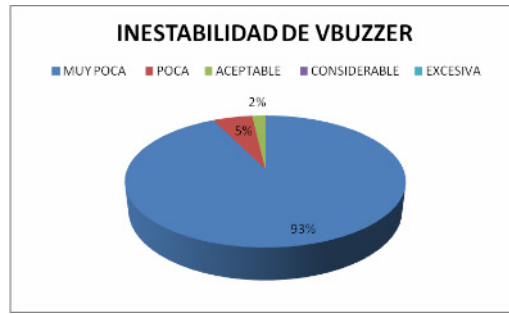


Figura 4.4 Valoración del usuario de la inestabilidad de cada aplicación.

4.6.5 Utilidad.

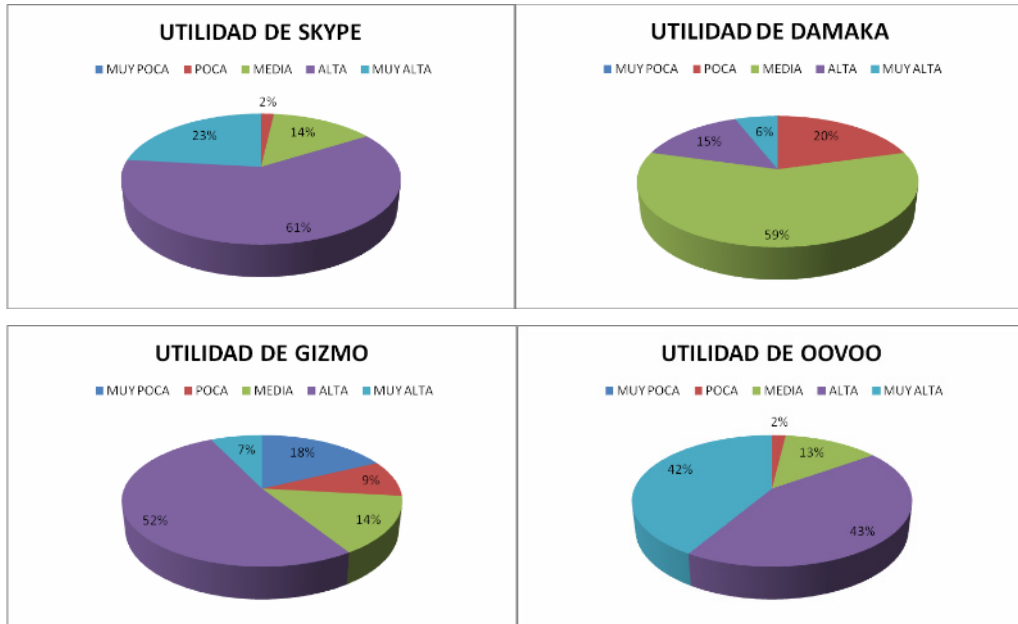
Las siguientes gráficas (Fig.4.5) muestran el nivel de utilidad que los encuestados consideran que puede tener cada aplicación.

La aplicación que consideran que tiene la mayor utilidad es ooVoo con un 42% le sigue Skype con un 23%.

La siguiente aplicación mejor considerada es Gizmo donde se califica con una alta utilidad con un 52%.

Damaka y Vbuzzer son las aplicaciones que se consideran que su utilidad es media con un 59% y 67%, respectivamente.

En Damaka también podemos obtener los peores valores ya que el 20% de los encuestados la considera como una aplicación con poca utilidad.



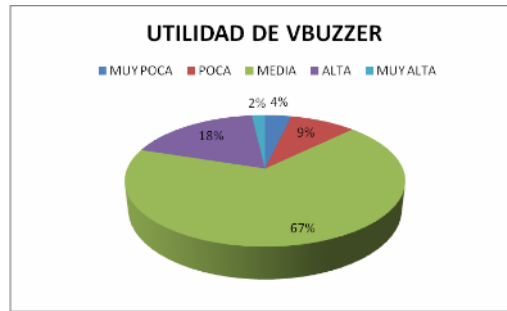


Figura 4.5 Valoración del usuario de la utilidad de cada aplicación.

4.6.6 Distorsión de la voz

En las siguientes figuras (Fig.4.6) se muestra el nivel de distorsión que los encuestados consideran que tiene cada aplicación.

En las cinco aplicaciones prevalecen los valores de poca o muy poca distorsión. No obstante los mejores valores para estos resultados son de 83% y 82% para la aplicación vBuzzer y ooVoo, respectivamente. Los sigue Damaka con un valor de 80%. Se contradice esta situación con el hecho de que se obtengan también en esta aplicación los peores valores, ya que existe un 17% de los encuestados que califica la distorsión de la voz como considerable.

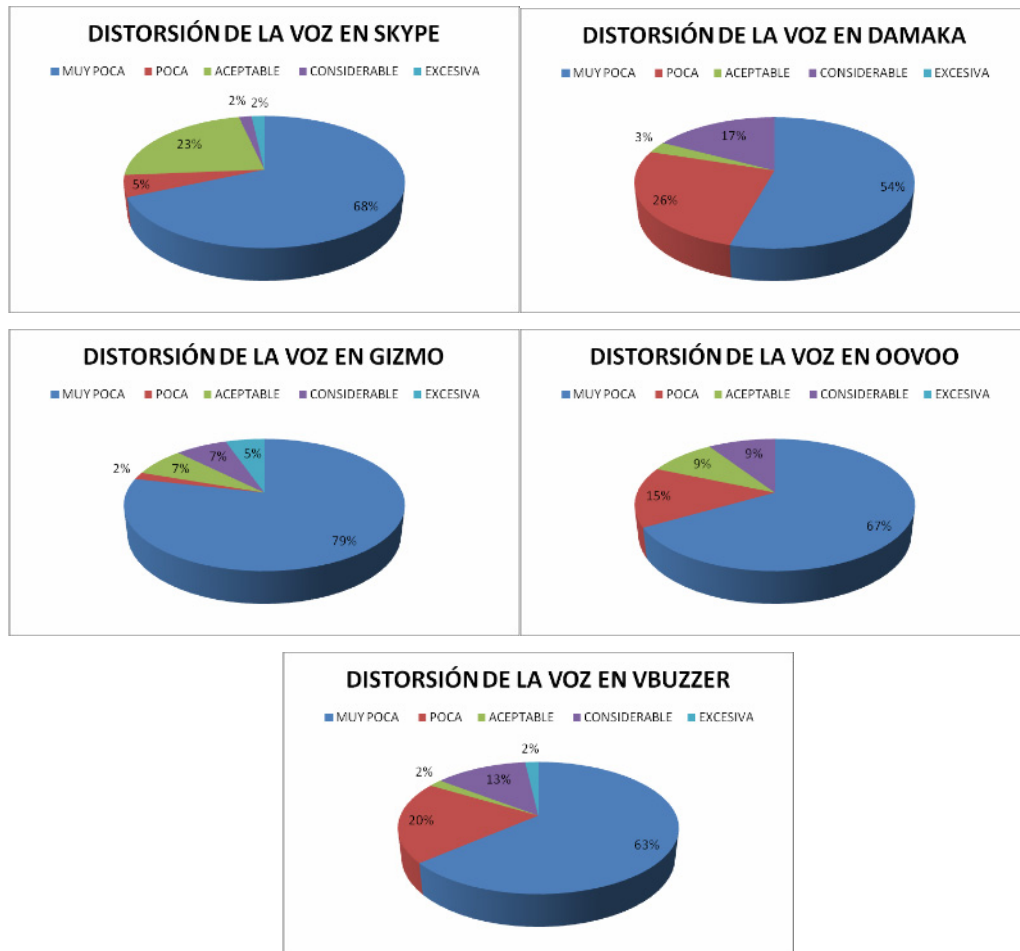


Figura 4.6 Valoración del usuario de la distorsión de la voz de cada aplicación.

4.6.7 Eco

Las siguientes gráficas (Fig.4.7) nos muestran el nivel de eco que los encuestados consideran que tiene cada aplicación.

Los valores obtenidos son muy buenos ya que todas las aplicaciones ofrecen muy poco nivel de eco. El 87% de los encuestados considera que ooVoo, Gizmo y vBuzzer son las aplicaciones que menor eco producen.

Los peores valores se obtienen en Damaka donde un 8% de los usuarios considera que el eco resultante es considerablemente importante, representa un nivel muy bajo frente al 70% que considera que el nivel de eco introducido en esta aplicación es muy bajo. Esto demuestra que la mayoría de los encuestados no percibe en las aplicaciones un importante nivel de eco en la comunicación que haga dificultoso seguir una conversación.

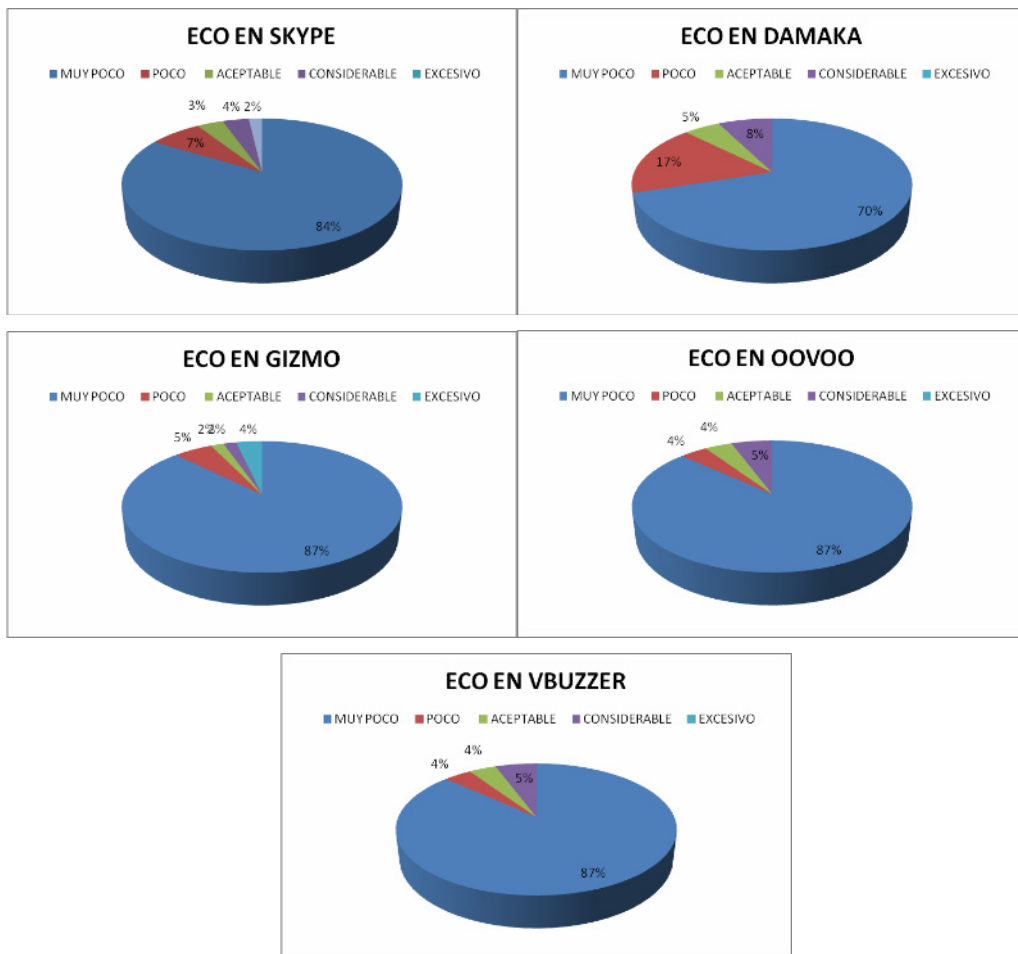


Figura 4.7 Valoración del usuario del eco de cada aplicación.

4.6.8 Ruido

Las siguientes gráficas (Fig.4.8) nos muestran el nivel de ruido que introducen las aplicaciones en la conversación.

Es destacable el nivel de ruido que introduce Damaka, el 40% de los encuestados considera que esta aplicación aporta un alto nivel de ruido a la conversación.

La aplicación según los encuestados que obtiene los valores más bajos para el ruido introducido es en ooVoo con un 66%.

En Skype y Gizmo el ruido introducido es muy poco o bajo para el 83% y 74%, respectivamente de los encuestados.

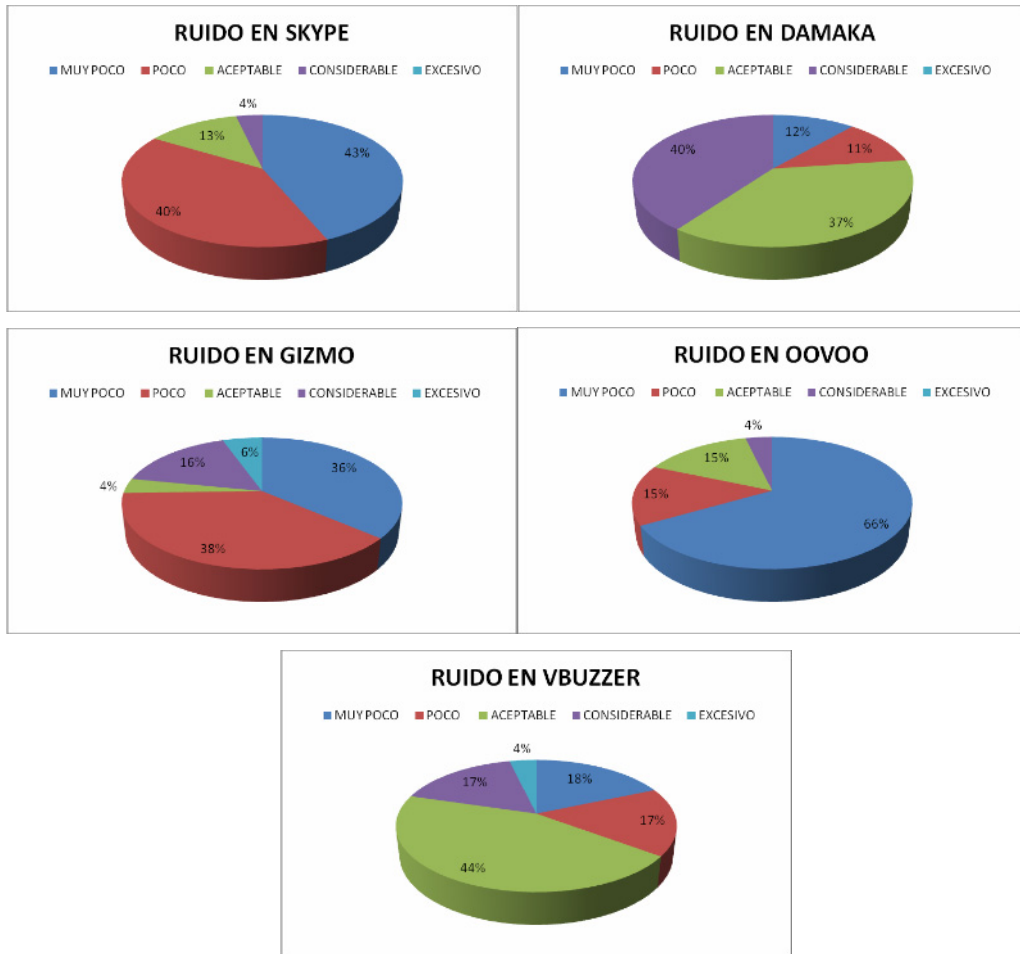


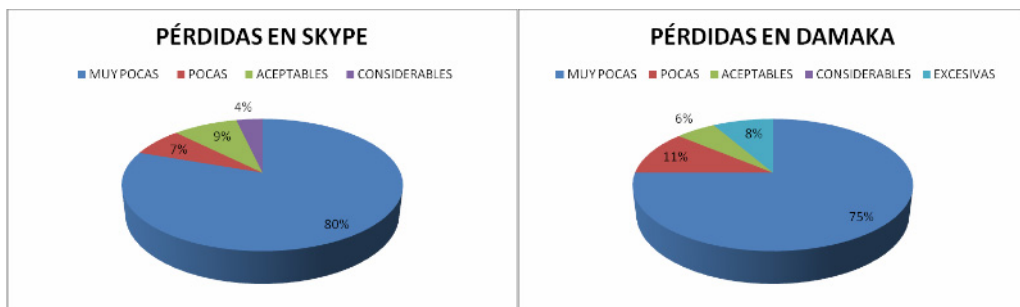
Figura 4.8 Valoración del usuario del ruido de cada aplicación.

4.6.9 Pérdidas.

A continuación (Fig.4.9) se muestran los porcentajes obtenidos para las pérdidas según los usuarios.

La aplicación que obtiene mayores pérdidas es Gizmo, un 12% de los encuestados consideró que en esta aplicación se perdían paquetes de voz mientras se comunicaban.

Los menores valores de pérdidas se obtienen para Skype donde el 80% de los usuarios considera que las pérdidas son mínimas.



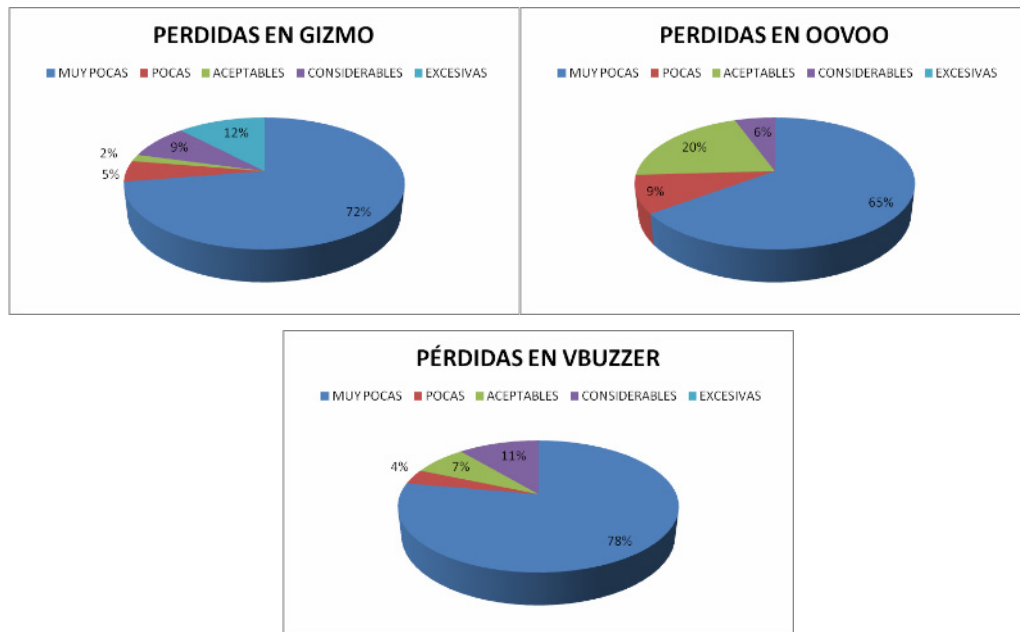


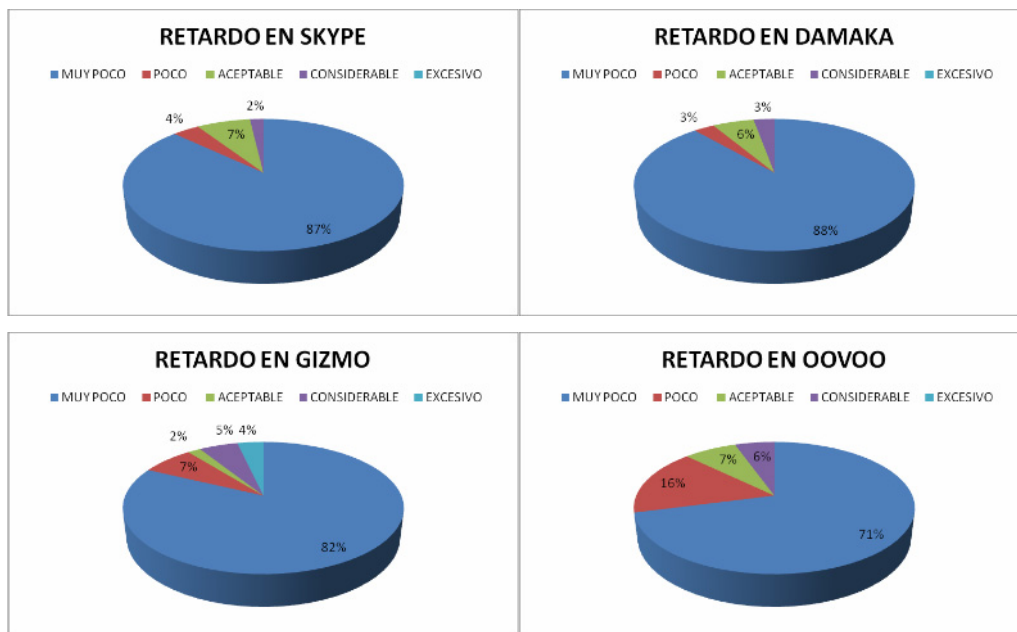
Figura 4.9 Valoración del usuario de las pérdidas de cada aplicación.

4.6.10 Retardo.

En la siguiente imagen (Fig.4.10) se muestra el retardo que los encuestados perciben en cada aplicación.

Cabe destacar que no existe ninguna aplicación con un alto nivel de retardos, es más, para las cinco aplicaciones el retardo que perciben los usuarios es bajo o muy bajo para más del 80% de las encuestas realizadas.

También cabe destacar que en la aplicación vBuzzer al igual que obtiene que el 81% de los encuestados considera que los retardos son bajos o muy bajos, también existe un porcentaje del 15% donde se obtiene el retardo considerablemente altos o excesivos.



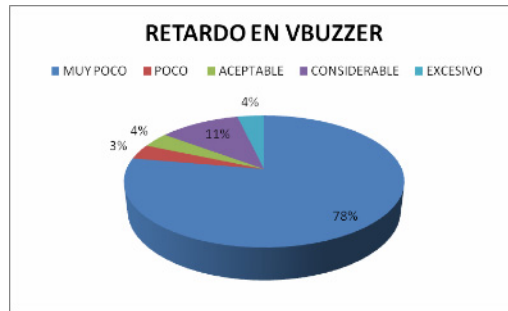


Figura 4.10 Valoración del usuario del retardo de cada aplicación.

4.6.11 Puntuación MOS

Vamos a mostrar la puntuación MOS obtenida para cada una de las cinco aplicaciones (Fig.4.11). En MOS se puntúa de 1 a 5, siendo 5 la nota más alta que se puede obtener.

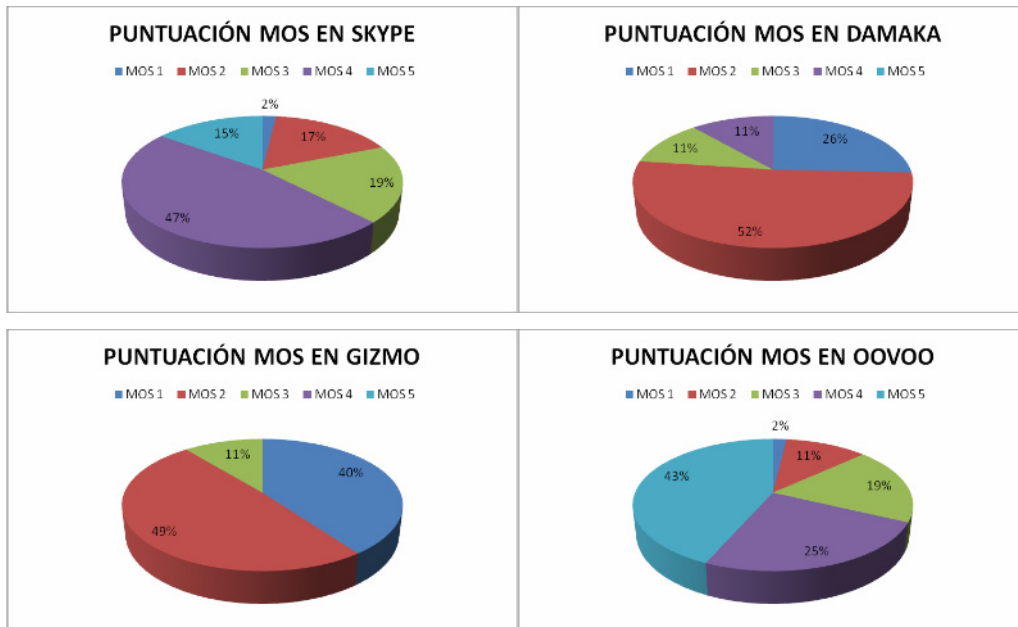
En Skype se obtiene que el 47% de los encuestados considera que esta aplicación tiene un 4 en la escala MOS. Seguida de un 19% que opinan que tiene un 3. Sólo el 15% de los encuestados le pone la máxima nota. La nota media MOS sería un 3,56.

La mayoría de los encuestados, un 52%, puntúa a Damaka con un 2. Seguida de un 26% que la calificaría con un 1. La nota media MOS sería un 2,07.

La aplicación Gizmo obtiene unos resultados muy bajos ya que el 49% de los encuestados considera que la puntuación para esta aplicación es 2 mientras que el 40% opina que es 1. Esta es la aplicación que obtiene los peores valores. La nota media MOS sería un 1,71.

La aplicación ooVoo es la que mejores resultados obtiene, debido a que el 43% de los encuestados la puntuaron con un 5 y el 25% con un 4. La nota media MOS sería un 3,96.

La aplicación vBuzzer obtiene que el 38% de los encuestados la califica con un 3, le sigue un porcentaje de 25% que corresponde a la nota de MOS 2 y un 24% de los encuestados la califica con un 1. La nota media MOS sería un 2,44.



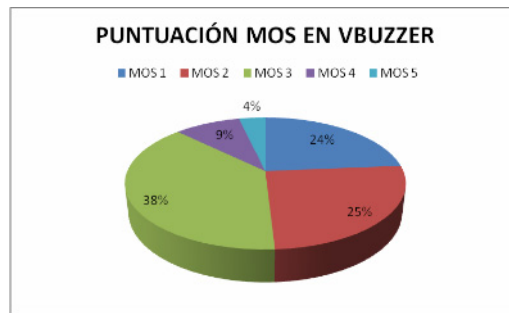


Figura 4.11 Valoración del usuario de la puntuación MOS de cada aplicación.

4.6.12 Calidad de video

La calidad de video solo se juzgo para cuatro aplicaciones (Fig.4.12) debido a que vBuzzer no ofrece este servicio.

La mejor calidad de video se obtiene para la aplicación ooVoo, en la cual el 72% de los encuestados considera que es excelente. La peor aplicación según los encuestados es Damaka donde con un 77% se califica la calidad de video como mala o pobre.

Gizmo es otra aplicación en la que la calidad es mala ya que más de la mitad de los encuestados lo considera así.

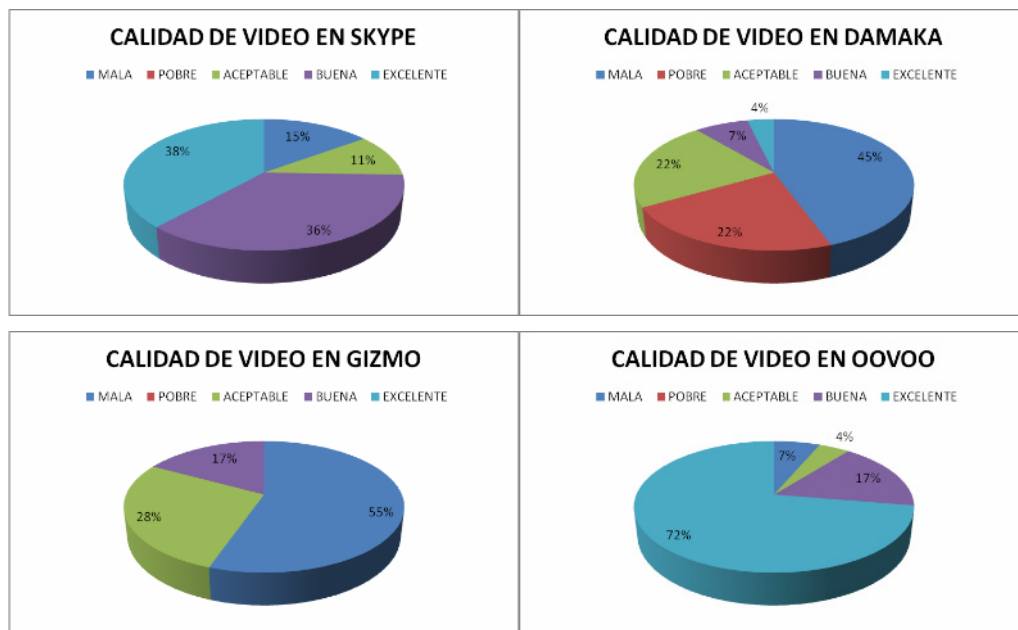


Figura 4.12 Valoración del usuario de la calidad de video de cada aplicación.

4.7 Comparativa QoS vs QoE

En esta sección vamos a establecer comparaciones entre los valores calculados para medir la calidad de servicio de las aplicaciones, como son tiempo entre paquetes, jitter, ancho de banda y paquetes perdidos, y los calculados para evaluar la calidad que el usuario experimentaba de la aplicación.

Para poder establecer comparaciones entre ambos primero vamos a establecer los resultados de algún trabajo anterior que muestre cuál es la relación entre ambos.

Seguidamente vamos a comparar la QoE y la QoS de cada aplicación por separado para ver de qué manera se ven afectadas las valoraciones de los usuarios en comparación con los valores cuantificados de los parámetros de la red.

4.7.1 Trabajos previos

En el año 2007, [PRAS07] presentó un trabajo de investigación que consistía en crear una aplicación que ajustaba los valores de los parámetros de la red a los que se obtienen mediante la QoE. Para poder crear esta aplicación con valores cercanos a los que en realidad experimenta el ser humano. Medían la QoE calificada por 21 personas sobre aplicaciones de voz y video en redes IP, los encuestados se dividían en tres grupos de distinta experiencia en este sector.

El experimento consistía en medir los parámetros de la red como son ancho de banda, retardo, pérdidas y jitter, en dos LAN totalmente aisladas. Para ello, instalaron una herramienta capaz de medir esos parámetros en los servidores de cada red. Como conclusión percibieron que la QoE del usuario variaba dependiendo del nivel de experiencia que el usuario tuviera con la tecnología VoIP. Los resultados obtenidos fueron que al aumentar las pérdidas y el jitter de la red disminuía la calidad experimentada por los usuarios de forma lineal hasta un límite inferior en el que se mantenía constante.

También se observó que la velocidad y magnitud de decrecimiento en la puntuación MOS obtenida era mayor cuando incrementaban los valores de pérdidas en la red que con la disminución del ancho de banda de la red.

4.7.2 Efectos de los parámetros medidos

A continuación explicamos los efectos que provocarían los diversos fallos o los valores indeseados de los parámetros de la red, y añadimos una tabla que nos muestra los valores óptimos para todos estos parámetros (véase Fig.4.13):

- Tiempo entre paquetes: si el tiempo entre paquetes es alto afecta a la fluidez de la conversación, ya que llegarían los paquetes muy separados entre sí y existiría cortes en las palabras o una entre frases del mismo interlocutor, se produciría distorsión en la voz. Si el tiempo que existe entre un paquete de VoIP y el siguiente es elevado eso nos muestra que hay mucha carga en la red, el retraso punto a punto aumentaría pudiendo provocar eco en la comunicación si este retraso es superior a 50 ms. También afecta directamente al jitter, porque este muestra la diferencia que existe entre los tiempos de llegada de paquetes consecutivos en el transmisor con los del receptor.
- Jitter: un alto nivel de jitter provoca que la conversación esté entrecortada, es decir que las palabras vayan llegando por sílabas y no se perciban como una palabra total, sino como trozos. Este parámetro al igual que el anterior también provocaría distorsión en la voz. Un alto nivel de jitter significaría que la red está congestionada, esto implicaría que las colas de salida de los *routers* se podrían colapsar y tendrían que tirar paquetes, lo que provocaría un aumento en la pérdida de paquetes. Podemos concluir que el jitter y el porcentaje de paquetes perdidos están relacionado.
- Ancho de banda: dedicar un alto ancho de banda para la transmisión de los paquetes de voz y video IP ayuda a conseguir una buena calidad en la comunicación. Sin embargo una baja cantidad de ancho de banda significa que existirían servicios de la aplicación que no podrán ser utilizados, por

ejemplo el video será de mala calidad o incluso nulo. La QoE refleja este parámetro en las puntuaciones MOS ofrecidas por los usuarios, debido a que un bajo ancho de banda refleja una mala calidad de los servicios ofrecidos que percibe el usuario. También se vería reflejado en la calidad de la voz y el vídeo a la llegada al receptor y en la distorsión que se produciría en recepción.

- Paquetes perdidos: un alto porcentaje de pérdidas provocaría vacíos en la conversación y palabras inteligibles, la conversación no sería entendible, estaría distorsionada. Si el número de paquetes perdidos fuera excesivamente alto no se podría realizar ni comunicación ya que ninguno de los dos extremos escucharía al otro, no le llegarían suficientes paquetes para que se pudiera reconstruir la conversación.

Factores de la red	Nivel Bueno	Nivel aceptable	Nivel Pobre
Ancho de banda	(>922] Kbps	(576-922) Kbps	[0-576) Kbps
Retardo	[0-150) ms	(150-300) ms	(>300] ms
Pérdidas	[0-1,5) %	(1,5-3) %	(>3] %
Jitter	[0-20) ms	(20-50) ms	(>50] ms

Figura 4.13 Valoración de los resultados de los parámetros de la red calculados

4.7.3 Comparación de resultados de QoS y QoE

Vamos a analizar paso a paso los valores obtenidos en la QoS para ir comparándolos con los que las personas encuestadas han calificado, QoE. Por tanto iremos deduciendo unas principales conclusiones para cada aplicación.

Debido a que las encuestas sólo se realizaron durante la mañana y la tarde, omitiremos relacionar la QoE obtenida con la QoS calculada durante la noche.

Si calculamos entre los 100 valores de MOS obtenidos en las encuestas de QoE, la media nos sale como resultado:

- ooVoo tiene un 3,96
- Skype tiene un 3,56
- vBuzzer tiene un 2,44
- Damaka tiene un 2,07
- Gizmo tiene un 1,71

Según la puntuación MOS con la que los encuestados han valorado las aplicaciones indica que la mejor para los usuarios es ooVoo seguida de Skype y la peor aplicación es Gizmo.

Si nos fijamos en la QoS obtenida para estas aplicaciones obtenemos los siguientes resultados:

- El tiempo que existe entre paquetes obtiene mayor valor para Gizmo, el más bajo es para Skype y ooVoo se encuentra en medio de ambos, sin embargo en algunas capturas llega a superar los valores ofrecidos por Gizmo.

- Ancho de banda: adquiere valores mayores para Skype y ooVoo siendo para el primero los más altos, sin embargo para Gizmo el ancho de banda utilizado es el más bajo con respecto a las cinco aplicaciones estudiadas.
- Jitter: obtiene los mayores valores para Gizmo, mientras que para Skype y ooVoo se obtuvieron los menores.
- Paquetes perdidos: Gizmo obtiene mayor porcentaje de pérdidas, ooVoo se encuentra en medio (sólo en una ocasión llega a superar el 3% de pérdidas), y Skype obtiene los menores resultados.

Gizmo

La aplicación Gizmo hemos comprobado en la QoE que es una de las aplicaciones que peor puntuación MOS ha obtenido (1,59). Vamos a analizar sus resultados medios de las encuestas sobre una puntuación máxima de 5 para cada característica valorada:

- La distorsión de la voz está entre muy poca y poca: 1,57.
- El eco está calificado como muy poco y poco: 1,35.
- El ruido en Gizmo se encuentra entre poco y aceptable: 2,13.
- Las pérdidas son pocas: 1,84.
- El retardo está entre poco y muy poco: 1,42.
- La puntuación media MOS obtenida es 1,71.
- La calidad de video es pobre: 2,07.

En los cálculos de QoS que hicimos nos sale como resultado:

- El tiempo entre paquetes es el más alto, ofrece resultados que se encuentran entre 60 ms y 100 ms.
- El ancho de banda utilizado es muy pequeño no llega a superar los 500 kbps.
- El jitter medio es alto durante la tarde que llega a sobrepasar los 20 ms recomendables. Durante la mañana no llega a 10 ms.
- El porcentaje de paquetes perdidos es mayor que el recomendable de 3%, durante la mañana está sobre el 10% y durante la tarde la mayoría de capturas ofrecen un 30%.

Vemos que los resultados hallados con la QoS son peores que los que las personas han valorado en su encuesta. Este hecho muestra la similitud del estudio realizado en Ohio donde en condiciones pésimas de la red las valoraciones experimentadas por los usuarios no lo eran tanto, se mantenían constantes en un límite inferior sin llegar a la mínima puntuación.

En la QoE las pérdidas encontradas están en una media de pocas, un 21% de los encuestados considera que existen pérdidas en la comunicación. Sin embargo mediante la QoS los resultados obtenidos son otros, todas las capturas se encuentran en valores altos que hacen pensar que la comunicación es muy difícil y el porcentaje de pérdidas en la QoE sería mucho más alto que el obtenido.

El jitter es alto durante la tarde que llega a sobrepasar los 20 ms recomendables, esto provocaría en la QoE obtenida valores de distorsión, retardo en la red o pérdidas elevados. Antes hemos visto que el 21% de los encuestados considera que existen pérdidas. Sin embargo sólo un 9% de los encuestados considera que existen demasiados retrasos o son excesivos para establecer una comunicación. Y un 12% de ellos consideran que la aplicación Gizmo provoca distorsión en la voz y en el video.

El ancho de banda calculado para hallar la QoS es muy bajo, es el menor valor de todas las aplicaciones. Este hecho se tiene que mostrar en la disminución de la calidad de voz y vídeo, una baja puntuación MOS, un aumento de las pérdidas y una gran distorsión de la voz y video. La QoE obtenida afirma casi todos los valores esperados para estos parámetros ya que la calidad de vídeo está considerada como muy mala para el 55% de los encuestados, en MOS es la aplicación que peor nota tiene, un 1,71. Sin embargo no es muy alto el porcentaje de personas que consideran que existen pérdidas o distorsión de la comunicación, un 21% y 12% respectivamente.

Damaka

Primeramente vamos a analizar los valores medios hallados en las encuestas realizadas para medir la QoE. Todas las puntuaciones medias son obtenidas sobre un valor máximo sobre cinco:

- La distorsión de voz en media obtiene poca: 1,83.
- El eco obtiene una puntuación media entre muy poco y poco: 1,51.
- El ruido obtiene valores medios como aceptables: 3,05.
- Las pérdidas en Damaka tienen una puntuación media de 1,47 entre pocas y muy pocas.
- El retardo obtiene un valor medio entre muy poco y poco: 1,24.
- El valor MOS medio es de 2,07.
- La calidad media de video está valorada como pobre, 2.03.

En las mediciones de QoS que hicimos obtuvimos los siguientes valores:

- El tiempo entre paquetes es de los más bajos en torno a los 20 ms.
- El ancho de banda usado obtiene valores muy variados para las capturas de la mañana que oscilan entre los 50 Kbps y los 1500 Kbps, sin embargo por la tarde todas se mantienen constantes en 500 Kbps.
- Los valores obtenidos para el jitter medio son prácticamente nulos.
- El porcentaje de paquetes perdidos se encuentra entre las aplicaciones con menor valor en torno al 2%.

El valor de pérdidas obtenido en la QoE se considera poco y se corresponde con las calculadas ya que no llegan a superar el 2%. Sólo el 8% de los encuestados considera que son excesivas.

El ancho de banda utilizado es muy pequeño, ya que mayoritariamente se encuentra entre 500 Kbps, esto hace pensar que la calidad de voz y video se vería afectada, por tanto aumentaría la distorsión en la comunicación, y también podría aumentar el porcentaje de pérdidas, sin embargo hemos visto que no sucede para todos estos parámetros. En la QoE la calidad de video tal y como preveíamos es

pobre. Sin embargo la distorsión en la voz es poca en media según los encuestados, sólo el 17% de los encuestados piensa que la distorsión es considerable y no pueden realizar una comunicación de buena calidad.

Los valores de jitter son nulos, lo que ayuda a entender que el porcentaje de pérdidas sea bajo.

El tiempo entre paquetes es muy bajo provocaría que en la QoE el retardo en la red y el eco obtuvieran valores pequeños, al igual que el porcentaje de pérdidas y la distorsión de la voz. Esto se cumple para todos estos parámetros medidos en la QoE que son muy bajos en media.

La distorsión de la voz se ve afectada por todos los parámetros medidos en la QoS, ya que aumenta debido al bajo ancho de banda utilizado y disminuye con los otros parámetros debido a los bajos valores que obtienen jitter, tiempo entre paquetes y pérdidas. Debido a estas diferencias encontramos valores muy diferentes en la QoE que valora este parámetro, aunque en valor medio obtiene muy poca distorsión.

A partir de los parámetros vistos tanto en QoS como en QoE, esta aplicación debería tener un valor MOS alto, sin embargo sólo obtiene un valor de 2.07, esto se debe a que otras características de la red que no hemos calculado le afectan en mayor medida. Además del poco ancho de banda utilizado para la comunicación, al servicio que más afecta es al video, también tenemos que tener en cuenta otros parámetros que no hemos medido en QoS pero que el usuario sí percibe y valora en la encuesta para determinar QoE. Estos parámetros son el ruido, donde el 40% de los encuestados considera que es importante, y consideraciones externas a la propia llamada pero que caracterizan la aplicación como pueden ser el tiempo que tarda en conectarse, el 48% de los encuestados considera que es excesivo, y la interfaz gráfica que ofrece que está calificada como pobre por el 46% de las personas.

Skype

Para la aplicación Skype los valores medios obtenidos para la QoE realizada nos muestra los siguientes datos, todos están puntuados sobre un valor máximo de 5.

- La distorsión de la voz tiene un valor medio entre muy poca y poca: 1,65.
- Obtiene muy poco eco en media con un valor de 1,33.
- El ruido medio se encuentra entre muy poco y poco con 1,78.
- Las pérdidas medias en Skype son muy pocas o pocas: 1,37.
- El retardo en media en Skype es muy poco: 1,27.
- La puntuación MOS obtenida es 3,56.
- La calidad de video obtenida es buena con un 3,82.

En el estudio realizado de QoS obtuvimos los siguientes valores para Skype.

- El tiempo medio entre paquetes era el menor de las cinco aplicaciones con un valor de 15 ms.
- El ancho de banda durante la mañana se mantiene sobre los 375 kbps, sin embargo durante la tarde es la aplicación que más ancho de banda dedica a la transmisión de VoIP con un valor en torno a los 1800 kbps.
- El jitter es prácticamente nulo.

- El porcentaje de paquetes perdidos es el menor de todas las aplicaciones con valores que no llegan a superar el 1%.

El nivel de pérdidas que los encuestados puntuaron a la hora de medir la QoE es muy bajo, este dato se corresponde con los valores obtenidos para el estudio de la calidad donde se obtenían los menores valores de todas las aplicaciones al igual que ocurre con las encuestas. Sólo el 4% de los encuestados consideraron que las pérdidas que se producían en Skype eran considerables.

El jitter es prácticamente nulo mediante el cálculo de QoS esto tendría que influir en unas bajas pérdidas, que como hemos visto se corresponde.

El valor bajo de tiempo entre paquetes en las mediciones de QoS tiene que tener como consecuencia para los encuestados que perciban un bajo retardo en la red y muy poco eco. Estos valores se corresponden con los obtenidos en la QoE y ya que sólo el 2% de los encuestados considera que Skype tiene un considerable retardo. Un poco superior, un 6% considera que la aplicación Skype aporta eco con valores importantes.

En el cálculo de QoS obtuvimos que la cantidad de ancho de banda que Skype utiliza aunque no es muy grande durante la mañana, sí lo es durante la tarde. Esto propicia que la calidad de voz y video sea alta, la puntuación MOS también se tiene que ver incrementada y las pérdidas serían bajas. Estas últimas ya hemos visto anteriormente que se corresponde la QoE con la QoS. Para los otros dos parámetros también se cumplen los valores esperados, ya que la calidad de video obtiene un valor de 3,82 sobre 5. La puntuación MOS es 3,56, aunque es alta, cabe pensar que esta aplicación está poco valorada por los encuestados en comparación con los buenos resultados de QoS que presenta que son los mejores de todas las aplicaciones y con los buenos valores con los que han puntuado el resto de características.

Esto se puede deber a que es la aplicación más conocida de todas las utilizadas, conlleva a que el encuestado espere lo máximo de esta aplicación y cualquier pequeño fallo que experimente lo puntuará negativamente con mayor dureza.

ooVoo

Los parámetros que los encuestados han medido están valorados en media y puntuados sobre un máximo de 5 de la siguiente forma:

- La distorsión de la voz tiene un valor medio entre muy poco y poco: 1,6.
- El valor de eco en ooVoo tiene un valor medio de muy poco con 1,27.
- El ruido en ooVoo tienen valor comprendido entre muy poco y poco: 1,57.
- Las pérdidas adquieren un valor medio de 1,67, entre muy poco y poco.
- El retardo obtiene un valor medio de 1,48, entre muy poco y poco.
- La puntuación MOS obtenida es la más alta con un valor medio de 3,96.
- La calidad de video experimentada es de 4,47.

La calidad de servicio calculada muestra los siguientes valores:

- Para el tiempo entre paquetes el resultado obtenido es alto, aproximadamente vale 60 ms. No es un valor muy bueno.

- El ancho de banda utilizado es un valor variable pero está en torno a los 1100 kbps. Un valor alto, suficiente para transportar audio y video de buena calidad.
- El jitter es prácticamente nulo.
- El porcentaje de paquetes perdidos durante la mañana se encuentra entre el 3% y 4%, sin embargo por la tarde estos valores disminuyen, al 1% y 2%.

El porcentaje de paquetes perdidos calculado es bajo, esto se corresponde con la puntuación que el usuario le aplica ya que sólo el 6% de los encuestados considera que las pérdidas que realiza ooVoo son considerables.

El jitter calculado prácticamente nulo está directamente relacionado con la probabilidad de pérdidas, que ya hemos comprobado que es muy baja.

El tiempo entre paquetes puede afectar si es muy alto al retardo de la red y al eco. OoVoo ofrece valores sobre los 60 ms, es uno de los valores más altos de entre las cinco aplicaciones, esto debería afectar al retardo de la red, al eco y a la distorsión, aumentándolos. Sin embargo los valores obtenidos para estos parámetros son bajos, no se ven afectados por el alto valor de tiempo entre paquetes. Sólo el 5% de los encuestados considera que ooVoo introduce un eco considerable. Un nivel más alto un 9% considera que la distorsión producida en la comunicación es considerable. Y un 15% considera que existe un retardo a tener en cuenta en la red.

El ancho de banda es un valor alto, esto provoca una buena transmisión de voz y vídeo simultáneamente, de esta manera la calidad de los servicios ofrecidos tendría que tener un alto valor de calidad de video, también de puntuación MOS y las pérdidas se deberían puntuar como bajas, ya que el ancho de banda es muy alto y el jitter es bajo.

Aunque las características de la aplicación son buenas, si las comparamos con las de Skype son menores que las de ésta. Sin embargo, posee un valor MOS superior a ella. Esto está influenciado porque los encuestados también tienen en cuenta otros parámetros no cuantificados en nuestro estudio de calidad de servicio como son la interfaz gráfica usada y el ruido introducido que en ooVoo es ligeramente inferior a Skype. También afecta la forma de presentar el video que en ooVoo es novedosa en formato y más compleja que la de Skype. Además sorprendentemente, aunque utiliza menor ancho de banda es más nítido que el usado por Skype.

vBuzzer

Los valores medios obtenidos a partir de las encuestas y medidos sobre un valor máximo de 5 son:

- La distorsión en la voz está medida entre muy poca y poca: 1,6
- El eco tiene un valor de 1,27, es muy poco.
- El ruido obtiene un valor comprendido entre poco y aceptable: 2,72.
- Las pérdidas en vBuzzer tienen un valor entre muy pocas y pocas de 1,51.
- El retardo obtiene valor comprendido entre muy poco y poco de 1,6
- La puntuación MOS es de 2,44.

El valor de los parámetros medidos para calcular la QoS es:

- El tiempo entre paquetes para vBuzzer tiene valores comprendidos entre 20 y 30 ms. Es de las aplicaciones que aportan los menores valores y más constantes.
- El ancho de banda utilizado por vBuzzer para realizar la transmisión de paquetes de voz IP varía entre los 500 y 700 kbps, si esta aplicación transportara también video, este valor sería ínfimo para lo que sería requerido. Sin embargo este valor es mayor al necesario para transmitir voz, cuyo ancho de banda necesario está en torno a 64 kbps.
- El jitter máximo que alcanza vBuzzer es de 1 ms.
- El porcentaje de paquetes perdidos durante la mañana se encuentra entre 3% y 4%, no obstante durante la tarde disminuyen estos valores que se mantienen constantes en el 1%.

El porcentaje de paquetes perdidos calculado llega a valer como máximo un 4%, aunque sobrepasa en pequeña medida los valores óptimos, son buenos, y eso se muestra en las encuestas realizadas, ya que obtiene como nota media que existen pocas pérdidas, sólo el 11% de los encuestados considera que las pérdidas son considerables.

El jitter calculado es un valor muy pequeño, no lleva a superar 1 ms, esto supone que el usuario tendría que experimentar que la aplicación aporta bajas pérdidas y como hemos visto anteriormente esto es así.

El tiempo entre paquetes obtiene valores entre 20 y 30 ms, es ligeramente superior al recomendado, sin embargo se encuentra en el ecuador de los tiempos hallados para las cinco aplicaciones. Esto debería provocar que el usuario evaluara un valor de eco pequeño, al igual que el retardo de la red y la distorsión de la voz.

Como podemos percibir en las encuestas realizadas se obtienen valores para retardo bajos en media, sólo un 15% de los encuestados considera que existe un retardo que pueda entorpecer la comunicación, el eco adquiere también valores bajos, ya que sólo el 5% de los encuestados considera que el eco es considerable y la distorsión de la voz obtiene valores similares ya que es el 15% de los encuestados los que consideran que es alta la distorsión introducida por vBuzzer.

El ancho de banda utilizado es grande, es nueve veces superior al recomendado para transmitir simplemente voz. Esta característica propiciaría que las pérdidas sean bajas, la distorsión de la voz sea mínima y la puntuación obtenida MOS sea alta. Anteriormente ya hemos visto que los parámetros de distorsión y pérdidas son los esperados para esta aplicación. Sin embargo la puntuación MOS es inferior a la que se podría esperar de esta aplicación que tiene las características estudiadas para la QoS con unos parámetros buenos.

Esto sucede porque hay que tener en cuenta otras características no calculadas al medir la calidad de servicio, como son el ruido introducido, ya que el 21% de los encuestados considera que el ruido es importante y un 44% considera que existe pero que es aceptable para realizar la comunicación. Otro parámetro que también afecta a la QoE de los usuarios es la interfaz gráfica usada, ya que un 10% considera que es pobre y un 64% opina que es aceptable simplemente. También afecta el hecho que esta aplicación carezca de video, disminuir la cantidad de servicios ofrecidos a los usuarios, que repercute directamente en la calidad que el usuario experimenta.

Conclusiones

Como conclusiones generales para todas las aplicaciones podemos decir que el aumento del retardo de la red afecta menos a los usuarios para estas aplicaciones que la disminución del ancho de banda a diferencia del trabajo que realizaron en 2007 en la universidad de Ohio [PRAS07].

El parámetro de la red que más afecta a la calidad experimentada por el usuario es sin duda el ancho de banda, ya que independiente del jitter, del tiempo entre paquetes o de las pérdidas, las aplicaciones son más o menos calificadas dependiendo de si usan mayor o menor ancho de banda para transmitir VoIP.

Sin embargo, que una aplicación tenga unos buenos parámetros calculados para QoS no significa que los valores obtenidos de QoE sean buenos, ya que esta puntuación depende de más características de la aplicación. Los parámetros que los creadores de aplicaciones tendrían que cuidar más si su propósito es satisfacer a los usuarios tendrían que ser:

- El ancho de banda que sea grande para que pueda transmitir video y voz con menor probabilidad de error o pérdida y con mejor calidad.
- La interfaz gráfica usada, hacerla más vistosa y novedosa ya que el usuario valora la aplicación en su totalidad y se tiene más en cuenta la apariencia que ofrezca el programa que algunos de los parámetros de red que ofrece.
- El ruido introducido por la aplicación, ya que muchas aplicaciones son castigadas por este valor aunque el resto de características que posee sean óptimas.
- El tiempo que tarda en realizar una conexión, se tiene que cuidar que el tiempo empleado para autenticar al usuario sea mínimo.

Capítulo 5

Conclusiones

5.1 Conclusiones

Las conclusiones obtenidas tras la realización de las mediciones de QoS y de las valoraciones de QoE son las que detallamos a continuación.

En las mediciones de los parámetros de QoS obtuvimos unos resultados que nos reflejan que en general podemos definir a Skype como la mejor aplicación. Esta afirmación la establecemos debido que era la que mayor ancho de banda dedicaba a la transmisión de los paquetes de VoIP, en la existía menor tiempo entre paquetes, proporcionaba unos valores de jitter muy bajos y también ha sido la que ha ofrecido un menor porcentaje de pérdidas de paquetes.

Por el contrario nos encontramos como peor aplicación a Gizmo ya que era la que poseía los peores valores en todos los parámetros medidos.

Al realizar las encuestas y medir los resultados obtenidos, no se refleja gran diferencia entre las calificaciones puestas a Skype, vBuzzer y a Gizmo y las esperadas tras el análisis de QoS. Sin embargo si es sorprendente las valoraciones hechas a las otras dos aplicaciones. Comenzamos analizando Damaka, en las mediciones de QoS reflejaba unos resultados buenos, no tanto como los de Skype pero estaban dentro de los considerados como “buenos”. Sin embargo, las encuestas la puntúan como la segunda peor aplicación por detrás de Gizmo. La otra aplicación que nos ha sorprendido es ooVoo ya que los valores de QoS no están entre los mejores, por ejemplo el tiempo entre paquetes está fuera de los niveles recomendados. No obstante es la aplicación que posee las mejores valoraciones en las encuestas realizadas.

En el capítulo anterior determinamos que el parámetro de red que está directamente relacionado con la QoE medida es el ancho de banda. Dato que demuestra los valores obtenidos para Damaka y para ooVoo, ya que la primera aplicación usa menor ancho de banda que la segunda. No obstante el ancho de banda utilizado por ooVoo no es mayor que el usado por Skype, sin embargo es ooVoo la que está valorada con una mayor puntuación en la QoE.

Esto nos lleva a la conclusión de pensar que el ancho de banda utilizado por la aplicación es proporcional a las valoraciones positivas que el sujeto hace sobre la aplicación. Pero también es destacable lo que influyen otros parámetros en las mediciones de QoE como pueden ser la interfaz gráfica usada o el tiempo que tarda en conectarse. Estos parámetros no modifican los valores medidos para la QoS en la transmisión de la voz o el vídeo. Sin embargo si son valorados subjetivamente y en los resultados de las encuestas queda demostrado que existe una ligera dependencia con ellos.

Referencias

1.1 Referencias

- [AWK08] Tutorial awk. <http://www.vectorsite.net/tsawk_1.html#m1>
- [CISC02] Paul J. Fong, Eric Knipp, David Gray, Scott M. Harris, Larry Keefer, Jr., Charles Riley, Stuart Ruwet, Robert Thorstensen, Vincent Tillirson, Michael E. Flanagan, *“Configuring Cisco Voice Over IP”*, Editorial Cisco Press, 2002.
- [CISC07] Business Communications: Cost and Productivity Improvements <<http://www.cisco.com/en/US/products/sw/voicesw/index.html>>
- [CLU04] Monólogo, “La canción de los payasos”, El club de la comedia, 2004.
- [DAM07] Damaka. <<http://www.damaka.com/consumers/download.htm>>
- [GIZ08] Gizmo. <<http://gizmo-project.softonic.com/descargar>>
- [GIZMO08] Ventajas de Gizmo5 en teléfonos móviles. <<http://www.gizmo5.com/>>
- [IPDC] P. Tom Taylor, Pat R. Calhoun y Allan C. Rubens. “IPDC: Base Protocol”. <<http://tools.ietf.org/html/draft-taylor-ipdc-00>>. July 1998.
- [ITUG.107] ITU-T Recommendation G.107, “The E-model, a computational model for use in transmission planning”. <<http://www.itu.int>>
- [ITUG.711] ITU-T Recommendation G.711, “Pulse code modulation (PCM) of voice frequencies”. <<http://www.itu.int>>
- [ITUG.722] ITU-T Recommendation G.722, “7 kHz audio-coding within 64 kbit/s”. <<http://www.itu.int>>
- [ITUG.723.1] ITU-T Recommendation G.723.1, “Coding of analogue signals by methods other than PCM”. <<http://www.itu.int>>
- [ITUG.729] ITU-T Recommendation G.729, “Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)”. <<http://www.itu.int>>
- [ITUH.245] ITU-T Recommendation H.245, “Control protocol for multimedia communication” <<http://www.itu.int>>
- [ITUH.248] ITU-T Recommendation H.248, “Gateway control protocol”. <<http://www.itu.int>>
- [ITUH.323] ITU-T Recommendation H.323, “Packet-based multimedia communications systems”. <<http://www.itu.int>>
- [ITUP.800] ITU-T Recommendation P.800, “Methods for Subjective determination of Transmission”. <<http://www.itu.int>>
- [ITUP.862] ITU-T Recommendation P.862, “PESQ towards Measuring Speech Intelligibility with Vocoders”. <<http://www.itu.int>>
- [ITUPAM] ITU-T Recommendation G.991.2, “Pulse Amplitude Modulation”. <<http://www.itu.int>>
- [ITUPCM] ITU-T Recommendation G.711, “Pulse Code Modulation (PCM) of Voice Frequencies”. <<http://www.itu.int>>

- [JAB08] Ventajas de Jabbin de transferencia de archivos. <<http://www.jabbin.com/>>
- [JUNI07] Juniper Networks, "Voice over IP" <www.juniper.net>
- [OOV08] ooVoo <<http://oovoo.softonic.com/descargar>>
- [PFC1836] Proyecto fin de carrera. Pedro Sánchez Sánchez, "Estudio y desarrollo de soluciones VoIP para la conexión telefónica entre sedes internacionales de una misma empresa", dirigido por Cristina López Bravo. Biblioteca de Antigones de la UPCT, referencia PFC1836, 24 abril de 2008.
- [PRAS07] Prasad Calyam, Eylem Ekici, Chang-Gun Lee, Mark Haffner y Nathan Howes, "A 'GAP-Model' based Framework for Online VVoIP QoE Measurement", Journal of Communications and Networks (JCN), 2007.
- [RFC1035] P. Mockapetris, "Domain Names". November 1987.
- [RFC1633] R. Braden, D. Clark, S. Shenker "Integrated Services in the Internet Architecture", Junio 1994.
- [RFC1889] H. Schulzrinne, S. Casner, R. Frederick y V. Jacobson "RTP: A Transport Protocol for Real-Time Applications", January 1996.
- [RFC2205] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin "Resource Reservation Protocol", Septiembre 1997.
- [RFC2212] S. Shenker, C. Partridge y R. Guerin "Specification of Guaranteed Quality of Service", September 1997.
- [RFC2474] K. Nichols, S. Blake, F. Baker y D. Black "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers". December, 1998.
- [RFC2509] M. Engan, S. Casner y C. Bormann "IP Header Compression over PPP", February 1999.
- [RFC2705] M. Arango, A. Dugan, I. Elliott, C. Huitema y S. Pickett. "Media Gateway Control Protocol (MGCP) Version 1.0", October 1999.
- [RFC2740] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang y W. Weiss "An Architecture for Differentiated Services", December 1998
- [RFC2885] F. Cuervo, N. Greene, C. Huitema, A. Rayhan, B. Rosen y J. Segers, "Megaco Protocol version 0.8", August 2000.
- [RFC3032] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li y A. Conta "MPLS Label Stack Encoding", January 2001.
- [RFC3261] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley y E. Schooler. "SIP: Session Initiation Protocol", June 2002.
- [RFC3545] T. Koren, S. Casner, J. Geevarghese, B. Thompson y P. Urdí, "Enhanced Compressed RTP (CRTP) for Links with High Delay, Loss and Reordering". July 2003.
- [RFC3711] M. Baugher, D. McGrew, M. Naslund, E. Carrara y K. Norrman "The Secure Real-time Transport Protocol (SRTP)", March 2004.
- [RFC4553] A. Vainshtein y YJ. Stein "Structure-Agnostic Time Division Multiplexing (TDM)", June 2006.
- [RFC4566] M. Handley, V. Jacobson y C. Perkins "SDP: Session Description Protocol", July 2006.
- [RFC768] J. Postel, "User Datagram Protocol", August 1980.
- [RFC791] J. Postel, "Internet Protocol", September 1981.

- [RFC793] J. Postel, "Transmission Control Protocol", September 1981.
- [SGCP] Mauricio Arango, Christian Huitema, SGCP (*Simple Gateway Control Protocol*), Version 1.0, May 5, 1998. <<http://huitema.net/voip/sgcp-v1-0.html>>
- [SKY08] Skype. <<http://skype.en.softonic.com/download>>
- [SKYPE08] Ventajas de Skype. <<http://www.skype.com>>.
- [VBU08] vBuzzer. <<http://vbuzzer-messenger.softonic.com/descargar>>
- [WIR08] Wireshark. <<http://www.wireshark.org/>>
- [ZRTP] Phil Zimmermann, Jon Callas y Alan Johnston "ZRTP: Extensions to RTP", Marzo 2006.

Bibliografía

1.1 Bibliografía

AllVoIPSolution.com “VoIP Equipment”. <<http://www.allvoipsolution.com/category/voip-review/>> Último acceso 20/04/2008.

Asterisk. <<http://www.asterisk.org/>>. Último acceso 30/06/2008.

Ciena: the network specialist “Voice over IP”. <http://www.ciena.com/products/productsapps_voip.htm> Último acceso 21/07/2007.

Cisco, “Business Communications: Cost and Productivity Improvements”. <http://www.cisco.com/en/US/products/sw/voicesw/products_category_business_benefits.html> Último acceso 25/07/2008.

David Soldani, “Means and Methods for Collecting and Analyzing QoE Measurements in Wireless Networks”, International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06), pp. 531-535, 2006.

Ekiga. <<http://www.ekiga.org/>> Último acceso 16/04/2008.

Enrique Vázquez Gallo, José Ignacio Moreno Novella, “CARacterización de nuevos SERvicios de TELecomunicación desde el punto de vista de la calidad de servicio e impacto en las reglas de diseño de redes de próxima generación (NGN)”, Jornada de Seguimiento de Proyectos en Tecnologías de Servicios de la Sociedad de la Información, 2007. TSI2005-07306-C02. <<http://peterpan.uc3m.es/jspTSI2007/resumenes/TSI2005-07306-C02.pdf>>. Último acceso 14/08/2008.

Federal Communications Commission (FCC). <<http://www.voip911.gov/>>. Último acceso 02/04/08.

Federal Communications Commission (FCC). <<http://www.fcc.gov/>>. Último acceso 22/09/2007.

Greg Goebel, “An Awk Primer”, 2008. <<http://www.vectorsite.net/tsawk.html>> Último acceso 10/07/2008.

IneoQuest “Media Quality of Experience”. <<http://www.ineoquest.com/content24.html>> Último acceso 05/05/2008.

James Peters, Jonathan Davidson, “Voice over IP Fundamentals VoIP, Cisco Press 2000”, Editorial Cisco Press, 2000.

Jesús Alberto Vidal Cortés, “El lenguaje de programación AWK” Madrid, Febrero 2002.

John Bartlett, “Quality of Experience (QoE)”, Marzo 2008. <http://www.nojitter.com/blog/archives/2008/03/quality_of_expe.html> Último acceso 05/05/2008.

Jonathan Davidson, James Peters, Manoj Bhatia, Satish Kalidindi, Sudipto Mukherjee, “Voice over IP Fundamentals, Second Edition” Editorial Cisco Press, 2006.

Mukundan Venkataraman, Shamik Sengupta, Mainak Chatterjee, “Towards a Video QoE Definition in Converged Networks”, Proceedings of the Second International Conference on Digital Telecommunications, 2007.

- Olivier Hersent, Jean-Pierre Petit, David Gurle, "Beyond VoIP Protocols", Editorial John Wiley & Sons Ltd, 2005
- Ourania Markaki, Dimitris Charilas, Dimitris Nikitopoulos, "Enhancing Quality of Experience in next Generation Networks Through Network Selection Mechanisms", IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1-5, 2007.
- Salomón del mundo VoIP. <<http://voip-mundo.blogspot.com/2008/03/anue-sistemas-patners-spirent.html>>. Último acceso 14/08/2008.
- Stefan Winkler, "Quality of Experience (QoE): An Important Measure of Success for IP-based Video Services", Editorial Symmetricom, Inc., 2007.
- Thomas Porter, Brian Baskin, Antonio Rosela, Choon Shim Andy Zmolek, "Practical VoIP Security", Editorial Syngress Publishing, Inc. 2006
- Tim Rahrer, Nortel Riccardo Fiandra, FastWeb Steven Wright, BellSouth "Triple-play Services Quality of Experience (QoE) Requirements", DSL Forum, Diciembre 2006. <www.dslforum.org/techwork/tr/TR-126.pdf> Último acceso 19/05/2008.
- UIT-T P.862 , "Serie P: Calidad de transmisión telefónica, instalaciones telefónicas y redes locales", Febrero 2001.
- Ulf Lamping, Richard Sharpe, Ed Warnicke "Wireshark user guide". <<http://www.wireshark.org/docs/>>. Último acceso 10/07/2008.
- Wikipedia. <<http://www.wikipedia.org>>. Último acceso 28/07/2008.