

**UNIVERSIDAD
AUTONOMA DE MADRID
ESCUELA POLITECNICA SUPERIOR**



TRABAJO DE FIN DE GRADO

**DESARROLLO DE UN SISTEMA DE MEDICIÓN,
MONITORIZACIÓN Y GESTIÓN DE REDES IPTV**

Diego Hernando Loeda
(diego.hernando@estudiante.uam.es)

JUN 2013

**DESARROLLO DE UN SISTEMA DE MEDICIÓN,
MONITORIZACIÓN Y GESTIÓN DE REDES IPTV**

AUTOR: Diego Hernando Loeda
TUTOR: Jorge E. López de Vergara Méndez

High Performance Computing and Networking
Dpto. de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
JUN 2013

Resumen — La continua evolución y globalización de la tecnología y las comunicaciones implica una adaptación de sus contenidos a las necesidades de los usuarios, así nace la nueva generación de conexiones sobre IP, como son Televisión sobre redes IP(IPTV), Telefonía sobre IP(VoIP), televisión en dispositivos móviles. Los nuevos servicios proporcionados a través de la red plantean una serie de retos a los proveedores. Necesitan saber si sus servicios se están proporcionando correctamente a los usuarios finales. Sin embargo, los parámetros utilizados actualmente para medir la calidad del servicio (ancho de banda, pérdida de paquetes, retrasos en tiempos de llegada, etc.) no muestran de una forma clara la calidad que los usuarios finales experimentan. Por lo tanto, necesitamos poder traducir la calidad de servicio a una calidad de la experiencia del usuario. Existen una serie de modelos para poder realizar este propósito, como utilizar la pérdida de paquetes para estimar la calidad experimentada en redes IPTV. Desafortunadamente, este modelo se aproxima a la calidad de experiencia media, pero no tiene un comportamiento deseado. Para resolver este problema, se desarrolla una aplicación que profundiza en el análisis del flujo recibido, que se adentra en la codificación de MPEG sobre RTP, para analizar las pérdidas en los distintos tipos de tramas transmitidas. Las pérdidas en estos tipos de tramas afectan de distinta manera a la calidad final, ya que unas transmiten mayor cantidad de información que otras. Previamente, se crea una gran base de datos de vídeos con pérdidas simuladas. Teniendo toda esta información, se define un modelo que traduce las pérdidas de los distintos tipos de tramas MPEG en una aproximación de la calidad experimentada por el usuario final. Los resultados demuestran que dicho modelo mejora la predicción frente a una predicción que sólo tiene en cuenta las pérdidas totales del flujo.

Palabras Clave — Calidad de experiencia, calidad de servicio, calidad de vídeo, VQM, pérdida de paquetes, puntuación subjetiva media, múltiple regresión lineal, tramas MPEG.

Abstract — The continuous evolution and globalization of technology and communication involves an adaptation of its contents to the needs of users. Thus was born the new generation of IP networks, such as Television over IP(IPTV), Voice over IP(VoIP), mobile television. These new services provided posed several challenges to providers. They need to know if their services are being provided properly. However, the parameters used actually to measure the quality of service (bandwidth, packet loss, delay, jitter, etc.) cannot be translated easily to quality of experience. Therefore, we need to create a function that transforms quality of service to quality of experience. There are several methods to achieve this goal, such as using packet loss rate(PLR) to estimate quality of experience in IPTV networks. Unfortunately, this model is able to predict an average quality of experience, but not fits well to the actual quality. To solve this problem, we develop an application that deepens the analysis of the received flow. This application delves into the encoding of MPEG over RTP to analyze losses of the different types of frames transmitted. A loss on these different types of frames has a different impact on the final quality, because has a different amount of information inside them and has different compression rates. Before building the model, we create a large database of videos transmitted on a simulated network with different losses. Knowing these losses, the model is able to predict the QoE. The final results show that this model has less mean square error and fit better to the real quality than the packet loss rate model.

Index Terms — Quality of Experience, Quality of Service, video quality, VQM, packet loss, Mean Opinion Score, multiple lineal regression, MPEG frames.

Agradecimientos

Es fácil entender que este apartado no se puede redactar hasta que no se ha dado por finalizado el trabajo realizado en este documento. Justo en este momento es cuándo te puede dar cuenta de cuánto tienes que agradecer a la gente que ha estado a tu lado a lo largo de este trabajo. Es complicado, hay muy poco espacio para todo lo que hay que agradecer a la gente que ha hecho posible esta etapa haya llegado a su fin.

A Jorge, el tutor de este TFG, por toda su ayuda y comprensión con mi dominio del inglés, tanto en este proyecto como en la publicación del artículo. A David y a Felipe, por todo el trabajo previo, ayuda y comprensión en este gran proyecto. A Paco, por haber sido mi tutor a lo largo de la carrera y por solucionarme cualquier problema relacionada con ella. Al grupo de investigación HPCN y a todos sus participantes por haberme dejado ser uno más, por el ambiente y por su apoyo a lo largo de este año y medio.

A Rafael, por su ayuda con la interfaz web, por todo su apoyo y amistad a lo largo de estos años de carrera. A todos los compañeros de la carrera, por estar ahí y conseguir sacar una sonrisa incluso en los días que no tienen fin. A todos los profesores que se han esforzado en transmitirnos sus valiosos conocimientos, de un modo u otro, a lo largo de la carrera.

A mis padres, por enseñarme a luchar por lo que se desea. A la familia y a toda la gente que ha estado conmigo a lo largo de la carrera por haber conseguido hacer la carrera más llevadera y hacer posible el sueño de finalizarla. Finalmente, a Luci por haber estado a mi lado día a día consiguiendo hacer esta etapa mucho más amena demostrándome todo su apoyo y comprensión a lo largo de tanto tiempo.

A todas las personas, que aunque no aparezcan aquí directamente, han estado presentes a lo largo de estos cuatro años y finalmente a lo largo de este trabajo, les debo mi eterno agradecimiento.

Índice general

1	Introducción	1
1.1	Objetivos	3
1.2	Fases de realización	3
1.3	Estructura del documento	4
2	Estado del arte	5
2.1	Introducción	5
2.2	Tipos de algoritmos de medición de calidad	5
2.3	Trabajo previo	6
2.4	Conclusiones	8
3	Análisis del problema	9
3.1	Introducción	9
3.2	Análisis de requisitos	9
3.2.1	Requisitos funcionales	9
3.2.2	Requisitos no funcionales	11
3.3	Conclusiones	12
4	Desarrollo de la solución	13
4.1	Introducción	13
4.2	División en módulos	13
4.2.1	Monitor de flujos	13
4.2.2	Funciones auxiliares	15
4.2.3	Analizador de capturas	16
4.2.4	<i>pcap2ts</i>	19
4.2.5	Gestor de comunicación y cálculo de calidad	20
4.2.6	Funciones auxiliares	20
4.3	Conclusiones	21
5	Creación del modelo	23
5.1	Introducción	23
5.2	Creación de la base de datos	23
5.3	Ajuste del modelo	26
5.3.1	Ajuste sobre PLR	26
5.3.2	Tipos de cuadros MPEG	27
5.3.3	Ajuste sobre IBP	28
5.4	Rendimiento	31
5.5	Conclusiones	33

6 Integración	35
6.1 Introducción	35
6.2 Proyecto PRINCE	35
6.3 Arquitectura	36
6.4 Sonda	37
6.5 Conclusiones	37
7 Conclusiones y Trabajo futuro	39
7.1 Introducción	39
7.2 Conclusiones	39
7.3 Trabajo Futuro	40
A Artículo publicado	43
B Interfaz web del sistema de medidas	55

Índice de cuadros

2.1	Comparativa algoritmos Full Reference	7
4.1	Campos de la estructura	18
5.1	Resultados de la regresión lineal	26
5.2	Coefficientes y resultados de la regresión lineal IBP (1)	29
5.3	Coefficientes y resultados de la regresión lineal IBP (2)	30

Índice de figuras

2.1	Diferentes tipos de algoritmos de medición de calidad	6
4.1	Estructura Gestión Canales	14
4.2	Configuración Monitor Flujos	14
4.3	Configuración Canales	15
4.4	Estructura Comunicación	16
4.5	Configuración Analizador Trafico	17
4.6	Cabecera Pcap2ts	19
5.1	Modelo Gilbert-Elliot	24
5.2	Calidad subjetiva vs modelo VQM [1]	25
5.3	Modelo de creación de la base de datos	25
5.4	Modelo VQM MOS vs Modelo PLR	26
5.5	Uso de cuadros IBP	27
5.6	Primer modelo IBP	28
5.7	Primer modelo IBP	29
5.8	Segundo modelo IBP	30
5.9	Segundo modelo IBP	31
5.10	Prueba de rendimiento	32
5.11	Prueba de rendimiento HD	33
6.1	Arquitectura de la simulación	36
6.2	Página Web	36
B.1	Pantalla de acceso	56
B.2	Página de inicio	56
B.3	Registros almacenados	57
B.4	Gráficas	58
B.5	Gestión de usuarios	58

Glosario

- ALU Alcatel-Lucent
- CDTI Centro para el Desarrollo Tecnológico Industrial
- CIF,QCIF Formato utilizado para compatibilizar los diversos formatos de vídeo digital.
- Delay Métrica utilizada para medir cuánto tarda un bit en atravesar la red de punto a punto.
- FFmpeg Herramienta de línea de comandos para codificar vídeos de una formato a otro.
- Gnuplot Herramienta muy flexible, que es capaz de generar gráficas de funciones y datos.
- GOP Grupo de imágenes dentro de un *stream* de vídeo.
- IPTV Denominación de distribución por suscripción de señales de televisión o vídeo usando conexiones IP.
- IP Protocolo de comunicación clasificado en la capa de red según el modelo OSI.
- IPFIX Protocolo creado para estandarizar los datos exportados en redes IP.
- IPTV Servicio de distribución por suscriptor de señales de televisión o vídeo.
- IPNQSIS Iniciativa europea lanzada para el desarrollo de herramientas para motorización y gestión de servicios sobre IP.
- ITU-T Organismo internacional encargado de regular los estándares de telecomunicaciones.
- Jitter* Variabilidad temporal durante el envío de señales digitales.
- MOS Métrica utilizada en la calidad de experiencia en una escala de 1 a 5.
- MPEG Moving Picture Experts Group.
- PLR Porcentaje de pérdida de paquetes
- PRINCE Producto Industrial para la Gestión de Calidad de Experiencia.
- PSNR Relación entre la calidad máxima de una señal y la calidad de la señal corrupta.
- QoE Calidad de experiencia, se define como la aceptabilidad global de un servicio.
- QoS Calidad de servicio, se refiere a una serie de aspectos de la comunicaciones.
- RTP Real-time Transport Protocol, protocolo de nivel de sesión utilizado para la transmisión de información en tiempo real.
- R^2 Coeficiente de determinación del modelo, representa cómo se ajusta el modelo a los datos reales.

SEE Error estándar de estimación.

SOAP Protocolo estándar de comunicación a través de archivos XML.

TcpReplay Es una herramienta de línea de comandos para retransmitir una captura de red.

UDP Es un protocolo de comunicación clasificado en la capa de transporte según el modelo OSI

u_int64_t Tipo de dato que representa a un dato sin signo de 64 bits.

VLC Reproductor multimedia y *framework* multimedia libre y de código abierto.

VoIP Voz sobre IP.

VQM Aplicación utilizada para el cálculo de la calidad de experiencia.

WSDL Formato XML utilizado para describir servicios web.

1

Introducción

El reciente incremento de consumo de tecnologías y comunicaciones por gran parte de la población requiere una evolución de las mismas, para adaptarse a las necesidades de los usuarios finales. Así nace la nueva generación de redes IP (IPTV, VoIP, televisión en dispositivos móviles). Millones de personas utilizan actualmente estos servicios, así que los proveedores deben encontrar un medio para poder controlar que el servicio se proporciona correctamente al usuario final. Esta medida es de vital importancia, ya que el negocio dependerá de la calidad proporcionada a los clientes. Por lo tanto, los operadores necesitan métodos efectivos para medir la calidad experimentada por los clientes.

Esta incógnita en los proveedores de servicios ha generado que los métodos de estimación de la calidad de vídeo hayan sufrido grandes cambios en los últimos años. Actualmente, los métodos que utilizan el vídeo original para realizar la comparativa con el vídeo degradado son los que mejor estiman la calidad percibida por el usuario final. Sin embargo, estos métodos no están pensados para la monitorización, sino que deben utilizarse para la comprobación de las estimaciones generadas por los métodos que actúan sin referencia total del vídeo.

La mayoría de las mediciones actuales se basan en un modelo sin referencia, la medición de una serie de medidas de la red. Esta información es capaz de proporcionarnos directamente una calidad de servicio (QoS) [2]. Esta información es muy útil para diagnosticar el estado y rendimiento de los enlaces de los enlaces y poder actuar rápidamente en caso de mal funcionamiento. Sin embargo, esta calidad de servicio no se ajusta bien a la calidad final percibida por los usuarios, ya que por ejemplo si el códec utilizado para comprimir los datos realiza una compresión muy agresiva la calidad experimentada por el usuario final bajará, aunque la calidad de la red sea perfecta.

Para solucionar este problema se crea una nueva medida: la calidad de experiencia (QoE) [3]. Esta medida subjetiva medirá la calidad media percibida por los usuarios finales. En nuestro caso específico, televisión sobre IP, la degradación de la calidad en la red dependerá de una serie de factores [4]:

1. **Pixelado del vídeo:** Este es uno de los problemas que sucede con más frecuencia. Normalmente se debe a una insuficiencia del codificador al tratar los datos. El usuario lo percibe como un grupo de píxeles con el mismo color que no actualizan su color en un intervalo de tiempo.
2. **Difuminado:** Sucede cuando no hay bits necesarios para codificar todos los detalles de un vídeo.
3. **Congelado:** En ocasiones la visualización se queda congelada. Este efecto se experimenta cuando no se dispone de suficiente ancho de banda, aunque también puede ser ocasionado por la pérdida de paquetes, si se pierde una trama que describe un cuadro completo, el vídeo se congelará hasta que llegue el siguiente cuadro.
4. **Sacudidas:** Ocurre cuando el flujo de vídeo no es capaz de mantener la tasa a la que el vídeo está codificado. Las principales causas son la variación del jitter y la tasa de envío.
5. **Sin vídeo:** o pérdida de señal, se debe principalmente a la caída de la conexión entre el usuario o el servidor.
6. **Mala sincronización:** Se produce cuando el audio y el vídeo están mal sincronizados, generalmente por problemas de codificación o en el ancho de banda. Esta es uno de los problemas más complejos y molestos en la visualización.

En consecuencia, se crea una métrica para poder medir la degradación percibida por el usuario final: el **MOS**(**M**ean **O**pinion **S**core). El rango de valores del MOS es de 1 a 5. Cuanto más bajo sea el valor significa una peor calidad percibida y cuanto más alta mejor calidad. Esta métrica nace en el campo de la telefonía para obtener la perspectiva del usuario final de la calidad de la comunicación. Rápidamente se estandariza en la recomendación de ITU-T P.800. Esta métrica se adaptó a la medición de calidad de vídeo, generando dos grupos de métodos de medición.

Los métodos contenidos en el primer grupo consisten en un grupo de sujetos de prueba que evalúan la calidad de los vídeos presentados. Este método es muy preciso, pero se necesitan muchos recursos para desarrollarlo y es complicado llevarlo a una medición de tiempo real. La ITU-T especifica una serie de métodos para estandarizar estas revisiones en la recomendación B.500.

Por otro lado, el segundo grupo de métodos se basan en una evaluación objetiva de la calidad del vídeo. Estos métodos se basan en modelos matemáticos que se aproximan a las mediciones subjetivas. En este caso la ITU-T también proporcionó la recomendación J.247 donde se especificaban una serie de directrices sobre la selección de métodos para la evaluación de la calidad de vídeo de forma objetiva cuando se dispone de la señal original de forma completa.

A lo largo de este trabajo se tratará de crear un método que sea capaz de medir la calidad del vídeo analizado de forma objetiva, eficiente y con un bajo coste.

1.1 Objetivos

La primera cuestión que promovió el desarrollo de este Trabajo de Fin de Grado fue la idea de crear una aplicación capaz de realizar unas medidas de calidad de experiencia del usuario final con un error en las mediciones aceptable.

Además, la creación del proyecto IPNQSIS, una iniciativa lanzada por Celtic y EUREKA, potenció el interés en la creación de este TFG. Esta iniciativa intenta crear un sistema de medición, monitorización y gestión de servicios de la nueva generación de conexiones IP. En España se creó un consorcio de empresas llamado Prince, en el que se participará a través de la empresa Naudit.

El problema es que ya existen muchos métodos para conseguir esta medición, como ya veremos en el capítulo 2. Estos métodos, desarrollados por empresas o grupos de investigación, tienen una complejidad matemática alta y son difíciles de mejorar. Pero tienen un principal problema, tienen un coste computacional muy alto y/o necesitan disponer del vídeo original como referencia para extraer la calidad del vídeo degradado.

Por lo tanto, la principal idea de este TFG evoluciona hasta llegar a crear una aplicación que sea capaz de medir la calidad, pero que debe premiar también la eficiencia, la portabilidad y, sobre todo, el bajo coste, para poder instalarla en dispositivos baratos y distribuirlo a lo largo y ancho de la red.

Además, si el modelo se distribuye por la red necesitará un sistema por el cual comunicarse con los servidores centrales cada cierto tiempo, para no crear sobrecarga en la red. Por lo tanto, los principales objetivos de este TFG son:

- Crear un modelo que sea capaz de medir la calidad de servicio experimentada por los usuarios finales con porcentaje de error reducido.
- Conseguir un código eficiente, portable y que no necesite una gran cantidad de recursos. Para ello deberemos crear un código que tenga poco procesamiento, sobre todo en el modelo.
- Crear un módulo de comunicación para poder exportar las estadísticas a un servidor externo. De este modo, se podrá introducir en la red mientras los servidores principales recaban información del estado de la red y de la calidad percibida.

1.2 Fases de realización

Para el desarrollo de este proyecto se han seguido una serie de fases, ya que un proyecto medio o mayor sin una organización clara durante el proceso está destinado al fracaso. La primera de todas ellas fue la investigación del estado del arte, para poder tomar ideas sobre los proyectos ya terminados y comprobar que no existiese nada parecido.

La segunda fase del proyecto trató sobre la investigación y realización de la codificación necesaria para crear una base de datos lo suficientemente grande como para entrenar el modelo.

La tercera fase del proyecto constó de un análisis del problema a un nivel más técnico, la división en módulos del sistema y un planteamiento de cómo se iba a efectuar la comunicación entre los módulos. Acto seguido se comenzó a desarrollar para posteriormente comprobar su correcto funcionamiento en las pruebas de integración.

En este último punto el esfuerzo invertido en el proyecto se dedicó a crear la base de datos, crear el modelo final y contrastar los resultados obtenidos. Además, paralelamente poco a poco se fue integrando el código en sistema de medición de trazas reales para comprobar su funcionamiento.

1.3 Estructura del documento

En el siguiente capítulo se realizará una exposición del Estado del arte, que ofrecerá una visión de los trabajos realizados previamente y así poderlos relacionar con el tratado en este documento.

En el capítulo 3 se realizará un análisis más técnico de la solución que se muestra en este documento.

En el capítulo 4 se explicará cómo se ha diseñado y desarrollado la aplicación, así como las estructuras creadas para comunicar todos los módulos.

En el capítulo 5 se explicará cómo se ha realizado el modelo necesario para poder calcular las estadísticas con los datos de la red previamente extraídos y se validará el trabajo mostrando su uso en un proyecto internacional.

En el capítulo 6 se describirá el proceso de integración del código en un entorno real.

Finalmente, en el capítulo 7 se verán las conclusiones del trabajo realizado, así como las aportaciones futuras que se pueden añadir al trabajo presentado.

2

Estado del arte

2.1 Introducción

En este capítulo se introducirán los tipos de algoritmos de medición existentes y se presentará el estado del arte actual. Así se podrá tener una visión más global del estado de las mediciones de calidad de experiencia existentes. Además se tomarán ideas de los trabajos existentes, que concuerden con los objetivos deseados de la aplicación, para evitar reinventar la rueda, y se observarán los fallos o el modo de adaptar su solución al problema planteado.

2.2 Tipos de algoritmos de medición de calidad

Como se ha mostrado anteriormente, una evaluación objetiva es mucho más económica y sencilla que una subjetiva. Como consecuencia de ello, nacieron tres grandes grupos de algoritmos objetivos de evaluación de calidad, divididos por la cantidad de información que toman del vídeo original [5]:

1. **FR(Full Reference)**: Como hemos comentado anteriormente, este grupo de algoritmos son los más precisos ya que tienen a su disposición el vídeo original y el degradado, pero también son los más lentos porque suelen comparar los dos vídeos a un nivel muy bajo (incluso pixel a pixel).
2. **RR(Reduced Reference)**: Este grupo de algoritmos necesita una pequeña parte de la información del vídeo original. Obtienen peores resultados que los algoritmos FR pero es posible utilizarlos en tiempo real.
3. **NR(No Reference)**: Los algoritmos de este grupo no requieren ningún tipo de referencia del vídeo original, únicamente se obtienen datos de la red y de la codificación del vídeo para proporcionar la estimación de calidad. Obtienen resultados peores que los algoritmos RR, pero no requieren un ancho de banda adicional para poder enviar información del video original.

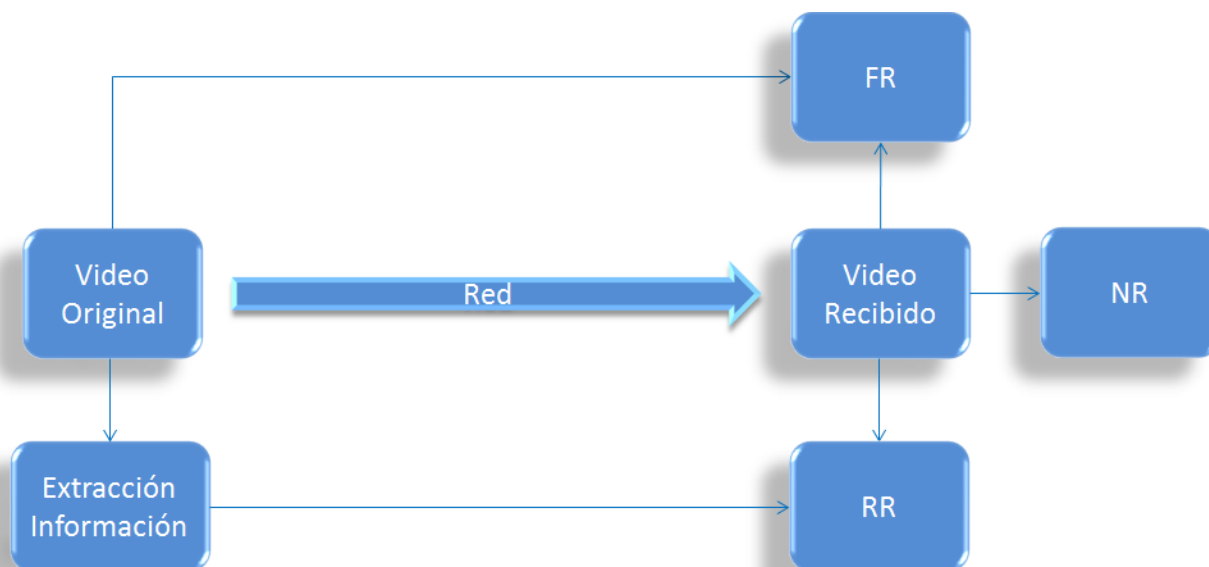


Figura 2.1: Diferentes tipos de algoritmos de medición de calidad

Como podemos imaginarnos un modelo FR no puede aplicarse en una situación real, ya que no podemos obtener el vídeo original en tiempo real. Aun así, este tipo de modelos son muy utilizados para comprobar el buen funcionamiento de los modelos restantes. Por otro lado, los modelos RR son una solución posible en tiempo real, pero consumen un ancho de banda que es bastante limitado en la mayor parte de los casos.

2.3 Trabajo previo

En este apartado se mostrarán las soluciones propuestas ya existentes al problema de la medición de la calidad de experiencia.

Como ejemplo de los algoritmos FR se presentarán los que recomienda la ITU:

1. **NTT** [6]: Este algoritmo realiza un alineado de alta precisión entre los dos vídeos. Posteriormente, se calculan las influencias visuales teniendo en cuenta la tasa de bit, la tasa de transferencia y la pérdida de paquetes. Este algoritmo utiliza PSNR para calcular el degradado del vídeo.
2. **PEVQ** [7]: La opción propuesta por *Opticom* es una de las más robustas de las actuales. Las principales características de este modelo son una rápida alineación temporal de los vídeos basada en una correlación multidimensional, además de la corrección de fragmentos congelados.
3. **Psytechnics** [8]: Este algoritmo tiene una gran capacidad para simular las respuestas humanas y lo consigue en tres pasos. El primero consta de un registro del video completo. A continuación, detecta los fallos perceptibles y finalmente calcula la calidad final del vídeo.

4. **Yonsei** [9]: Este algoritmo fue creado por un grupo de investigación. Principalmente se basa en la degradación de los bordes del vídeo, una de las características bastante sensible a la percepción humana. Su veredicto se basa la utilización del algoritmo PSNR entre los bordes de los distintos vídeos.

Principalmente la ITU recomienda el uso de los modelos de PEVQ o Psytechnics, pero por desgracia son de pago. Por otro lado los otros dos modelos, pero su punto débil es que se basan en el PSNR, que no tiene una buena correlación con la calidad del vídeo. A continuación se mostrarán otros algoritmos FR que se puedan utilizar como guía de referencia a la hora de fabricar el modelo:

1. **VQM** [1]: Este modelo se ha desarrollado por la **NTIA**(**N**ational **T**elecommunications and **I**nformation **A**dministration). Gracias a su resultado positivo en los test se tomó como estándar en **ANSI**(**A**merican **N**ational **S**tandards **I**ntitute), además de en **VQEG**(**V**ideo **Q**uality **E**xpert's **G**roup).
2. **MOVIE** [10]: Es un modelo desarrollado por el Laboratorio de Ingeniería de Imagen y Vídeo. Este modelo presenta tres vertientes: MOVIE index, Spatial MOVIE Index y Temporal MOVIE Index.

Las pruebas realizadas en el estudio de Seshadrinathan y su equipo [11] muestran la correlación de los distintos algoritmos. Podemos ver el extracto de los resultados en la Tabla 2.1.

Cuadro 2.1: Comparativa algoritmos Full Reference

Algoritmo	Wireless	IP	H.264	MPEG
VQM	0.721	0.638	0.652	0.781
MOVIE	0.810	0.715	0.766	0.773
S-MOVIE	0.792	0.704	0.706	0.691
T-MOVIE	0.811	0.719	0.779	0.817

En el caso estudiado, como vamos a trabajar con una codificación MPEG, se ha seleccionado el algoritmo VQM como referencia, ya que tiene un índice de correlación alto para ese caso específico.

Los algoritmos RR no se mostrarán ya que, como se ha expuesto antes, no son una solución 100 % válida por el consumo añadido de red. Por lo tanto, se presentarán una serie de ejemplo de algoritmos sin referencia para comprobar su metodología.

El trabajo de M. Venkataraman y M. Chatterjee [12] es un buen punto de partida para crear un algoritmo que no utilice ninguna referencia del vídeo original. Sin embargo el funcionamiento del algoritmo no es extremadamente portable. Esto es porque el algoritmo recoge una serie de parámetros de la red y realiza una búsqueda en su base de datos, lo que requiere una gran capacidad de almacenamiento y un cálculo forzoso al buscar el registro más parecido en su base si el exacto no se encuentra, lo que puede incrementar los costes finales del dispositivo físico.

El trabajo de Fuzheng Yang y Shuai Wan [13] también es muy interesante, ya que realizan un estudio de la variación de la calidad (en escala MOS) en función del ancho de banda. Por otro lado, este estudio no tiene un gran papel en IPTV, ya que el ancho de banda a utilizar no sufrirá unos cambios tan bruscos como para notar esos cambios de calidad. Como se puede apreciar, los vídeos utilizados en su estudio son los mismos que se utilizarán en este para estimar el MOS.

El trabajo desarrollado por P. de la Cruz Ramos, J. Navarro Salmerón, R. Pérez Leal y F. González Vidal [14] es un gran trabajo y una buena base para comenzar a crear el modelo deseado, pero este trabajo se estanca en un análisis de las pérdidas, da buenos resultados pero un análisis más exhaustivo puede dar mejores resultados. Por todo esto, el modelo auxiliar será parecido al que se desarrolló, pero se creará uno que indague en las pérdidas de la capa MPEG para mejorar la precisión.

El trabajo desarrollado en París por J. Pokhrel [15] es una buena guía de referencia, ya que también crear un algoritmo sin referencia, intentar crear una correlación entre parámetros de QoS con QoE a través de un sistema de reglas de inferencia difusas. Además, comparan los resultados obtenidos con un banco de pruebas y con el algoritmo VQM. El problema es que para nuestro modelo final, llevar la aplicación a una sonda de bajo coste, el algoritmo utilizado puede ser demasiado costoso y se deberá crear un modelo que consuma menos recursos.

Por último, el trabajo desarrollado en la Universidad de Würzburg [16] se basa exclusivamente en un análisis de la calidad de servicio frente a la tasa de pérdida de paquetes, en este trabajo se indagará más, entrando en la capa MPEG, para poder realizar estimaciones más precisas.

2.4 Conclusiones

En este capítulo se ha mostrado el estado del arte en la medición de calidad de vídeo. Se han valorado cada uno de los algoritmos existentes y se ha seleccionado un modelo a seguir para crear el propio y se ha observado los puntos negativos de los algoritmos existentes para el problema propuesto para evitarlos.

En el siguiente capítulo se tomarán las ideas expuestas en este capítulo y en los objetivos para resolver el problema y se desarrollará un análisis más a fondo el problema y plantear las soluciones necesarias para afrontarlo. Además de seleccionará la familia de algoritmos necesaria para solucionar el problema planteado.

3

Análisis del problema

3.1 Introducción

En este capítulo se analizará de una manera más técnica el problema de la medición de calidad de experiencia en redes IPTV. Extrayendo de los objetivos deseados las ideas principales y tomando la familia de algoritmos NR, que es la que más se adecua a nuestras necesidades. La solución que se propone no se detiene en tomar medidas de la red e intentar crear un método que intente relacionar la calidad de servicio y la calidad de experiencia.

El modelo que se expone en este documento se introduce en la codificación del vídeo para observar las pérdidas a nivel de la capa MPEG, para poder ver cómo afectan estas pérdidas a la calidad. Este método se entrenará con el algoritmo FR VQM, ya que se ha demostrado que tiene una buena correlación con la calidad real en MPEG.

Las ventajas de este método sobre los anteriores presentados son: el bajo consumo de recursos y unos resultados aceptables, lo que implica que podría llevarse a un dispositivo físico para medir la calidad de experiencia de los usuarios finales sin un coste demasiado elevado. Además se planteará la división de los módulos necesarios para crear el algoritmo final.

3.2 Análisis de requisitos

A continuación se mostrará una lista de los requisitos funcionales y no funcionales que deberá tener la aplicación. Se explicará la funcionalidad que deberá tener la versión final, para que no exista ningún cabo suelto.

3.2.1 Requisitos funcionales

En esta sección se explicarán los requisitos funcionales que deberá tener la aplicación una vez terminada, esto es las funcionalidades que deberá tener. Para facilitar la comprensión de todos los requisitos se ha planteado la aplicación como un conjunto de módulos. Cada módulo se encargará de un conjunto de funcionalidades comunes. Los módulos mencionados serán los siguientes:

- **Captura de canales:** Este módulo se dedicará a volcar el contenido útil de la red (los paquetes relacionados con la emisión de vídeo) en una serie de ficheros para su posterior análisis.
- **Análisis de la red:** Este módulo se encargará de recoger la información necesaria para el cálculo de la calidad de las capturas previamente realizadas.
- **Comunicación y cálculo de calidad:** Este módulo se encargará de calcular la calidad con los datos de la red y almacenarlos tanto en ficheros como enviarlos por la red hasta un servidor.

A continuación se detallarán los requisitos funcionales para cada uno de estos módulos del sistema de una forma técnica.

Captura de canales

RF1: Se deberán capturar y almacenar el contenido de la red para su posterior análisis.

RF1.1: Se dividirán las capturas en tamaños de ventana parametrizables.

RF1.2: Se organizarán las capturas en función de la fecha de captura.

RF2: Se filtrará el contenido de la red para solo almacenar contenido útil.

RF2.1: Se creará un filtro para almacenar únicamente los paquetes con contenido de audio o vídeo.

RF3: Se creará un gestor de canales activos e inactivos.

RF3.1: Se leerá la lista de canales desde un fichero y se comprobará su actividad.

RF4: Se deberá parametrizar el código para que sea altamente configurable.

RF4.1: Se leerá un fichero de configuración donde se encuentren las configuraciones que cambien con mayor frecuencia.

RF4.2: Se definirán dentro del código valores que serán fácilmente modificables.

Análisis de la red

RF5: Se analizarán las capturas a nivel de RTP (pérdidas, jitter, retrasos, etc.).

RF6: Se analizarán las capturas a nivel de MPEG (pérdidas de tipos de cuadros, pérdidas en la codificación, etc.).

RF7: Se almacenarán las capturas con pérdidas para su posterior análisis.

RF8: Se deberá parametrizar el código para que sea altamente configurable.

Comunicación y cálculo de calidad

RF9: Se calculará la calidad de experiencia con los datos de la red.

RF9.1: Se podrá calcular la calidad en función del análisis en nivel MPEG si el vídeo contenido está correctamente codificado.

RF9.2: Si la codificación no permite el análisis MPEG, se proporcionará una cálculo de la calidad basado en las pérdidas en RTP.

RF10: Se almacenarán los datos obtenidos en un sistema de ficheros.

RF11: Se comunicará cada cierto tiempo con el servidor para enviar la información de la calidad obtenida, así como el análisis realizado por el anterior módulo.

RF12: Se proporcionará un submódulo que gestione el espacio disponible en el disco.

RF13: Se deberá parametrizar el código para que sea altamente configurable.

3.2.2 Requisitos no funcionales

A continuación se mostrarán otro tipo de requisitos, los no funcionales. Estos requisitos son las características que debe cumplir nuestro sistema, pero no están relacionados con su funcionalidad. En el caso planteado primaremos:

RNF1 Rendimiento: La aplicación deberá tener gran redimiendo si se desea implantarla en dispositivos de bajo coste para medir la calidad en todos los hogares que se implante IPTV.

RNF2 Disponibilidad: La aplicación deberá estar analizando el tráfico constantemente, para poder solucionar un problema en la red en cuanto se produzca.

RNF3 Estabilidad: La aplicación deberá estar corriendo continuamente, por lo que debe ser altamente estable.

RNF4 Portabilidad: La aplicación debe poderse ejecutar en cualquier sistema Linux para que no dependa del dispositivo en el que se ejecute.

RNF6 Escalabilidad: Deberá ser altamente escalable a la hora de analizar varios canales a la vez, ya que así el error en la red se detectará mejor cuando se produzca.

RNF7 Mantenibilidad: Deberá ser un código limpio y bien organizado para poder mantener la aplicación con facilidad.

RNF7 Gestionabilidad: El código estará debidamente documentado y no será complejo su uso.

RNF8 Seguridad: Los datos almacenados estarán correctamente protegidos y no se permitirá su acceso sin autenticación.

RNF9 Transparencia: Los datos obtenidos no dependerán del canal a medir.

3.3 Conclusiones

En este capítulo se han mostrado las características deseables del proyecto y se ha mostrado la división de módulos del proyecto, haciéndolo más sencillo de analizar, desarrollar y mantener. Además se hizo especial hincapié, en crear una aplicación fácilmente configurable, ya que esto creará un mantenimiento y portabilidad mucho más sencillo. En el siguiente capítulo veremos cómo cada uno de los requisitos mostrados en el proyecto van tomando forma. Además se mantendrá la división en módulos haciendo referencia al lema algorítmico: *"Divide y vencerás"*.

4

Desarrollo de la solución

4.1 Introducción

En este capítulo se mostrará el diseño y el desarrollo de la solución propuesta en el anterior capítulo. Se tomarán las características deseadas del anterior capítulo para hacerlas poco a poco más tangibles. Además, se mantendrá el sistema de módulos creado en la fase de análisis para facilitar la comprensión, el diseño y desarrollo.

4.2 División en módulos

Como se ha especificado anteriormente, se dividirá el diseño y el desarrollo de la solución en módulos para facilitar el esfuerzo necesario para realizarlo y comprenderlo. Se especificarán las medidas necesarias para crear los módulos y crear una comunicación interna entre ellos.

4.2.1 Monitor de flujos

Como hemos visto en el anterior capítulo, el principal cometido de este módulo será la de gestionar los canales deseados y volcar el contenido útil para el módulo analizador. Por lo tanto deberemos crear una estructura para gestionar los canales deseados (comprobar su actividad y liberarlos si están inactivos). Para todo esto utilizaremos la estructura *IPSession* que podemos ver en la Figura 4.1.

Los campos de tipo *IPFlow* contienen información relativa al flujo (direcciones destino y origen, puertos, etc.), los campos de tipo *u_int64_t* están destinados a almacenar la información sobre el tiempo de recepción del primer y último paquete. Así como el tiempo de volcado del contenido. El campo *active_node* es utilizado como lista donde se almacena la información del canal monitorizado, para posteriormente volcarlo a fichero.

```

1  typedef struct IPSession
2  {
3      IPFlow incoming;
4      IPFlow outgoing;
5      IPFlow *actual_flow;
6      u_int64_t exportation_timestamp;
7      u_int64_t lastpacket_timestamp;
8      u_int64_t firstpacket_timestamp;
9
10     node_l *active_node;
11
12 }IPSession;

```

Figura 4.1: Código para estructura de gestión de canales

Con esta información podremos volcar las capturas de los canales de forma ordenada en un sistema de ficheros ordenados por el tiempo del primer paquete capturado del flujo. El contenido de los canales será volcado en una serie de ficheros de capturas de red (*.pcap) y serán almacenados según el fichero de configuración lo especifique. En la figura 4.2 podemos ver un ejemplo del fichero de configuración.

```

1  #directorios
2  datadir=data
3  capturedir=capture
4  bindir=bin
5  #ruta absoluta
6  lecturadir=/home/hpcn/demo_ipnqsis/ipnqsis_zador/lecture
7  #interfaz de captura
8  intf=eth0
9  file_lectura=file.pcap
10 #duracion de la traza de captura
11 capture_duration=10
12 #fichero configuracion canales
13 channels=canales.m3u
14 #fichero a crear para notificar la modificacion del fichero channels y pedir que el
   procesador lo recargue. Al hacerlo lo borra
15 channelsupdaterequest=channels.update
16 #cada cuantos segundos se chequea la existencia del fichero channelsupdaterequest
17 channelsupdateinterval=3600)

```

Figura 4.2: Configuración de Flow Process

Como podemos observar en el fichero de configuración, en las primeras líneas deberemos informar a la aplicación de la localización relativa de los directorios de datos, exportación de capturas y de código. Acto seguido deberemos especificarle la interfaz de lectura de canales y el fichero de lectura, en caso de que la interfaz no este disponible (diseñado para leer capturas de red). Finalmente, el tamaño de ventana deseado, el fichero de canales y el método para recargar la lista de canales. Acto seguido la aplicación accede al fichero de canales, como ejemplo se proporciona uno en la Figura 4.3.

```
1 #EXTINF:-1,[000] Canal Promocional
2 rtp://@239.11.73.81:16081
3 #EXTINF:-1,[001] Tve
4 rtp://@239.11.73.82:16082
5 #EXTINF:-1,[002] La 2
6 rtp://@239.11.73.83:16083
7 #EXTINF:-1,[003] Antena 3
8 rtp://@239.11.73.84:16084
9 #EXTINF:-1,[004] Cuatro
10 rtp://@239.11.73.85:16085
11 #EXTINF:-1,[005] Tele 5
12 rtp://@239.11.73.86:16086
13 #EXTINF:-1,[006] La sexta
14 rtp://@239.11.73.87:16087
```

Figura 4.3: *Canales.m3u*

Una vez concluido el submódulo de configuración, el módulo de captura inicializa las variables necesarias y lanza dos hilos. El primer hilo ejecuta la función *copyThread*, que veremos en la sección 4.2.2, que se encarga de volcar a fichero el contenido de los canales. El segundo hilo ejecuta la función *receptor_socket*, encargada de extraer estadísticas de la red, o la función *receptor_fichero* dependiendo de la configuración de la interfaz.

4.2.2 Funciones auxiliares

En este apartado trataremos con más detalle las funciones auxiliares que se invocan desde los hilos que se generan en el módulo de motorización de flujos y desde la propia función principal.

copyThread

CopyThread tiene como función final volcar el contenido de los canales a ficheros **.pcap*. Para ello comienza inicializando las variables y reservando la memoria necesaria. Acto seguido, utilizando un semáforo, con el que se asegura de ser el único proceso que está capturando los canales, comienza a comprobar los flujos que están actualmente activos y los que han ido expirando, para posteriormente eliminarlos. Finalmente desbloquea la captura de canales y llama a la función *dump_flows* para que pueda guardar el contenido de los flujos a fichero.

receptor_socket

La función auxiliar *receptor_socket* tiene como objetivo el preprocesamiento de los paquetes recibidos a través del socket inicializado previamente en la función principal utilizando la función *crearSocketRawRx*. Una vez extraída la información del tiempo de captura de cada fichero la función se encarga de actualizar la información disponible sobre los flujos y de actualizar el sistema de directorios en los que se va almacenando la información.

receptor_fichero

Esta función tiene el mismo funcionamiento que la anterior, pero tiene la diferencia de que los datos los toma del fichero previamente indicado en la configuración de la aplicación en el campo *file_lectura*.

crearSocketRawRx

Esta función se encarga de inicializar el *socket* en formato libre (para poder capturar la información desde la capa más baja del modelo OSI hasta el contenido del paquete RTP), que se utilizará para la captura de canales.

4.2.3 Analizador de capturas

El principal cometido de este módulo será recoger las capturas volcadas por el anterior módulo, abrirlas y aplicarles algún filtro adicional, si fuera necesario, y extraer la información necesaria para que el modelo pueda extraer la calidad. Para ello deberá extraer estadísticas a nivel RTP/UDP (tasa de pérdidas, jitter, retraso, etc.) y a nivel MPEG (pérdidas MPEG y de codificación). Para comunicar todas estas estadísticas con el módulo que calculará la calidad, se crea una estructura para facilitar la comunicación del sistema. En la figura 4.4 podemos ver el código de la estructura.

```

1  struct ns1_Ipfix
2  {
3      //Capa IP
4      int protocolIdentifier ;
5      //Capa UDP
6      LONG64 destinationIPv4Address ;
7      int destinationTransportPort ;
8      LONG64 sourceIPv4Address ;
9      int sourceTransportPort ;
10     //Ip de la sonda
11     LONG64 exporterIPv4Address ;
12     //Capa RTP
13     //Informacion
14     LONG64 collectionTimeMilliseconds ;
15     xsd_integer rtpFlowId ;
16     LONG64 rtpSSRC ;
17     LONG64 rtpSampleTime ;
18     xsd_integer rtpStartTime ;
19     int rtpMediaType ;
20     int rtpMOSCAlg ;
21     int rtpObservationType ;
22     //Estadicas
23     int rtpJitterCount ;
24     LONG64 rtpJitterSum ;
25     float rtpMinMOS ;
26     float rtpAvgMOS ;
27     LONG64 rtpPacketCount ;
28     LONG64 rtpPacketCountDiscarded ;
29     LONG64 rtpPacketCountLoss ;
30     //Capa MPEG
31     LONG64 mpegTsLoss ;
32     float iLoss; ;
33     float bLoss; ;
34     float pLoss; ;
35 };

```

Figura 4.4: Código para estructura de comunicación

Todos los campos utilizados en esta estructura han sido extraídos de IPFIX(IP Flow Information Export) [17] o de IETF(Internet Engineering Task Force) [18]. En la tabla 4.1 se explicará detalladamente el significado de los campos. Cabe añadir que algunos campos se han ampliado de tamaño al que se puede ver en la tabla, por problemas de compatibilidad con código Java del servidor de recepción de estadísticas mostrado en el capítulo 6.

Al intentar crear la conexión con el servicio web se encontró un problema. Se detectó que los datos de tipo *unsigned* no son bien traducidos a sus tipos de datos, por lo que hubo que ampliar el tamaño de todos los campos. Como no se puede ampliar en tamaño de un *u_int64_t* en un ordenador de 64 bits de una forma cómoda, este tipo de dato se introdujo en un *String* en hexadecimal ya que esos datos específicos son fácilmente utilizables en esa codificación también.

Una vez comprendido esto, se podrá comenzar a ver el funcionamiento del analizador de tráfico. Al comienzo, el módulo leerá el fichero de configuración del que extraerá la información necesaria para su correcto funcionamiento. En la figura 4.5 se puede ver un ejemplo de fichero de configuración.

```
1 10
2 60
3 1500
4 /home/hpcn/demo_ipnqsis/ipnqsis_zador/capture/
5 /home/hpcn/demo_ipnqsis/ipnqsis/pcap2ts/moved/
6 /home/hpcn/demo_ipnqsis/ipnqsis/pcap2ts/demonstrator_repetidas/
7 /home/hpcn/demo_ipnqsis/ipnqsis/pcap2ts/video/
8 /home/hpcn/demo_ipnqsis/aux_stat_files/
9 /home/hpcn/demo_ipnqsis/general_stats_results.dat
10 http://10.0.104.199:8080/PRINCE_Servidor/services/princeReceiverNaudit
```

Figura 4.5: Configuración de Pcap2ts

En este fichero se podrá configurar, según el orden de aparición:

- Longitud de ventana en segundos (duración de los ficheros de lectura recibidos)
- Intervalo de generación de estadísticas en segundos
- Tamaño del *buffer* a contemplar para pérdidas en milisegundos (utilizado para contemplar cuando se ha de descartar un paquete por recepción tardía)
- Directorio de lectura
- Directorio de almacenado de capturas con pérdidas
- Directorio almacenamiento información flujos
- Directorio almacenamiento vídeo decodificado
- Directorio de estadísticas auxiliares
- Fichero de estadísticas final
- Dirección del Servicio Web donde se almacenarán las estadísticas

Cuadro 4.1: Campos de la estructura

Campo	Capa	Bits	Explicación
protocolIdentifier	IP	32	Identificador del protocolo de la capa IP
destinationIPv4Address	UDP	64	Dirección IP versión 4 del destinatario del paquete
destinationTransportPort	UDP	32	Puerto del destinatario del paquete
sourceIPv4Address	UDP	64	Dirección IP versión 4 del que envía el paquete
sourceTransportPort	UDP	32	Puerto del que envía el paquete
exporterIPv4Address	UDP	64	Es la dirección IP versión 4 de la sonda que extrae el tráfico. Este campo es un buen identificador para saber qué sonda envía la información si disponemos una red de ellas
collectionTimeMilliseconds	RTP	64	Almacena el tiempo de recepción del primer paquete del flujo, que no tiene que ser el mismo que el de la captura
rtpFlowId	RTP	128	Identificador del flujo RTP
rtpSSRC	RTP	64	Almacena el valor del campo SSRC de la cabecera RTP
rtpSampleTime	RTP	64	Almacena el valor del tiempo de la cabecera RTP
rtpStartTime	RTP	128	Tiempo de inicio del primer paquete de la última captura procesada (puede servir para ver si un flujo sigue activo o no)
rtpMediaType	RTP	32	Almacena el valor del tipo de datos transportados de la cabecera RTP
rtpMOSCAlg	RTP	32	Es el identificador del algoritmo utilizado para el cálculo del MOS, los valores tomados han sido 128 para el primer modelo y 129 para el segundo
rtpObservationType	RTP	32	Informa del punto de observación del analizador, el valor asignado es 1 (la observación se realiza en el receptor)
rtpJitterCount	RTP	32	Conteo de los <i>jitters</i> sumados
rtpJitterSum	RTP	64	Suma de <i>jitters</i> del flujos
rtpMinMOS	RTP	32	MOS mínimo del flujo
rtpAvgMos	RTP	32	MOS medio del flujo
rtpPacketCount	RTP	64	Conteo de paquetes según el número de secuencia de RTP
rtpPacketCountDiscarded	RTP	64	Conteo de paquetes descartados por tardanza
rtpPacketCountLoss	RTP	64	Conteo de paquetes perdidos según el número de secuencia de RTP
MpegTsLoss	MPEG	64	Pérdidas dentro de un paquete RTP. Estas pérdidas pueden deberse a una sobrecarga en el codificador
iLoss	MPEG	32	Porcentaje de pérdidas de cuadros de tipo I
bLoss	MPEG	32	Porcentaje de pérdidas de cuadros de tipo B
pLoss	MPEG	32	Porcentaje de pérdidas de cuadros de tipo P

Una vez visto esto podemos observar el funcionamiento del módulo. Primero, la aplicación inicializará todas las variables necesarias y se almacenará localmente el contenido del fichero de configuración.

Acto seguido, el módulo se introducirá en un bucle infinito en el que se leerán las capturas existentes en el directorio de lectura, se ordenarán por orden de creación y se le proporcionarán una a una a la función *pcap2ts* para que extraiga la información necesaria para su análisis. Esta función puede devolver tres valores:

- '0' si no ha habido ningún problema, en cuyo caso el fichero se elimina y se almacenan sus estadísticas.
- '1' si se han encontrado pérdidas en la captura, en cuyo caso se almacenará el fichero y sus estadísticas.
- '-1' si no se ha podido procesar, en cuyo caso se almacena el error y se continúa analizando el resto de capturas.

Finalmente se le proporcionarán las estadísticas al último módulo de la aplicación a través de la estructura *ns1_ipfix* para que calcule la calidad de experiencia, indicándole en el campo *rtpMOSCAlg* que tipo de algoritmo tiene que utilizar, en función de si se han podido extraer las estadísticas de MPEG, o por el contrario si el vídeo no está correctamente codificado y no se han podido calcular esas estadísticas.

4.2.4 *pcap2ts*

La función *pcap2ts* es la encargada de procesar las capturas de los canales de IPTV para poder sacar las estadísticas necesarias. Para ello esta función requiere que se le proporcione la estructura de *config_info*, que contiene todo el contenido del fichero de configuración, los filtros que se quieran aplicar a la captura, el nombre de un fichero de captura y el registro resultado.

```
1 int pcap2ts(struct config_info *cfg, char * filter, char * pcap_filename, ipfix *ipfix_register){
```

Figura 4.6: Cabecera de *pcap2ts*

Una vez inicializadas todas las variables y el registro, aplicados los filtros, se abre para lectura el fichero **.pcap* y se comienza a procesar. Se recorre el fichero paquete a paquete y se va almacenando temporalmente la información de cada uno de los paquetes para poder ir calculando las pérdidas de paquetes, a través del número de secuencia de la capa RTP, los paquetes descartados por el buffer, la distancia en milisegundos entre paquetes y el jitter.

Posteriormente, se comprueban los errores percibidos en la capa MPEG, contabilizando tanto las pérdidas de fragmentos de tramas como las de tramas completas o sus cabeceras. Por último, se comprueba si existen errores en la captura y se devuelve la información.

4.2.5 Gestor de comunicación y cálculo de calidad

En este apartado analizaremos a fondo el diseño y desarrollo requerido para la creación del módulo de comunicación y de cálculo de la calidad de experiencia. Como se ha podido ver en el capítulo anterior capítulo este módulo se dedicará a la comunicación con el servicio web, para ello deberá tomar la dirección del servicio web del fichero de configuración visto en la Figura 4.5, así como el directorio donde se quieren generar las estadísticas de forma local.

Posteriormente deberá ir tomando uno a uno los registros generados por el módulo analizador, a través de la estructura proporcionada en la Figura 4.4 y calcular la calidad con la función *mos_model_plr* o la función *mos_model_ibp*, en función del valor proporcionado en el campo *rtp-MOSAlg*.

Una vez recogidos los valores de calidad, se calculan la calidad mínima y la media. Además, las estadísticas se van agrupando por flujos para facilitar su estudio posterior. También en este módulo se proporciona un método para ir descartando los flujos que no se utilicen, lo que disminuye notablemente los recursos que requiere la aplicación para operar correctamente.

Finalmente, cada vez que se cumpla el intervalo de tiempo se enviarán las estadísticas al servidor a través de la función *send_record* y se volcarán en un fichero local a través de la función *generate_general_stats*.

4.2.6 Funciones auxiliares

En este apartado se estudiarán las funciones auxiliares utilizadas por el módulo de cálculo y comunicación.

mos_model_plr

Esta función es la encargada de calcular la calidad de experiencia del usuario final en base a las pérdidas de paquetes producidas en una ventana de tiempo. Si no existen pérdidas esta función devuelve un 5 en la escala de MOS. Sin embargo, si existen perdidas la calidad se calcula como resultado de la fórmula 4.1.

$$MOS(plr) = a + b \cdot plr \quad (4.1)$$

Los coeficientes de la fórmula expuesta se explican en el capítulo 5.

mos_model_ibp

Esta función es la encargada de calcular la calidad de experiencia en función de las perdidas existentes en la capa MPEG producidas en una ventana de tiempo. Si no existen pérdidas esta función devuelve un 5 en la escala de MOS, pero si existen perdidas la calidad se calcula como resultado de la fórmula 4.2.

$$MOS(I_{loss}, B_{loss}, P_{loss}) = a + b \cdot I_{loss} + c \cdot B_{loss} + d \cdot P_{loss} \quad (4.2)$$

Los coeficientes de la fórmula expuesta se explican en el capítulo 5.

send_record

Esta función se encarga de enviar un grupo de registros (un flujo por registro) al servicio web, para ello es necesario proporcionarle la dirección de servicio y una estructura donde se informa del número de registros enviados y un array de ellos. Este código invoca una función que envía la información a través de un protocolo SOAP. Finalmente retorna la respuesta del servidor si no existe ningún problema.

generate_general_stats

Esta función se encarga de reunir la información de todos los registros volcados para volcarlos a fichero, donde se generará una entrada por cada flujo existente.

4.3 Conclusiones

En este capítulo se ha mostrado cómo se han desarrollado cada uno de los módulos necesarios para poder solucionar el problema planteado. Se han mostrado las estructuras necesarias para el funcionamiento y la comunicación existente entre ellas, así como la comunicación externa, pero queda una gran incógnita.

¿Cuáles son los coeficientes que se han de utilizar en los modelos para estimar la calidad de experiencia? En el capítulo 5 se mostrarán los pasos que se han seguido para conseguir estos coeficientes, así como las conclusiones obtenidas de ellos.

5

Creación del modelo

5.1 Introducción

En este capítulo se abordará la incógnita existente sobre los coeficientes de las formulas 4.1 y 4.2. Para ello se deberá crear una gran base de datos de vídeos y comprobar la validez de las formulas, así como los índices de error existentes en ellas y su ajuste al modelo real. Finalmente se extraerán las conclusiones sobre los coeficientes obtenidos y sobre la utilización de los distintos modelos.

5.2 Creación de la base de datos

Para la realización de los experimentos, se han tomado un conjunto de 23 vídeos proporcionados por Telecommunication Networks Group en [19]. Este conjunto de vídeos son altamente utilizados para este tipo de experimentos. Una de las principales razones por lo que son ampliamente utilizados es que tienen unos niveles de complejidad espacio-temporal muy diferentes. El conjunto de secuencias utilizado tienen una resolución CIF y QCIF a 25 *frames* por segundo. Usando este conjunto de vídeos, se podría comparar el trabajo con otros experimentos realizados.

Para la creación de la base de datos se ha realizado, para cada uno de los 23 vídeos codificados en H.264, una recodificación (gracias a la herramienta *ffmpeg* [20]) de su formato original a MPEG con cuadros del tipo que se quieren analizar. Posteriormente, se realiza una corrupción del flujo de datos a través de una tasa de pérdida de paquetes.

Para simular un modelo real, se han añadido pérdidas encadenadas siguiendo un modelo de do estados de Gilbert-Elliot [21]. Se han simulado seis diferentes tasas de pérdidas {0.1 %, 0.4 %, 1 %, 3 %, 5 %, 10 %}, con una ráfaga media de pérdidas de 3 paquetes, como es característico de las redes IP [22, 23].

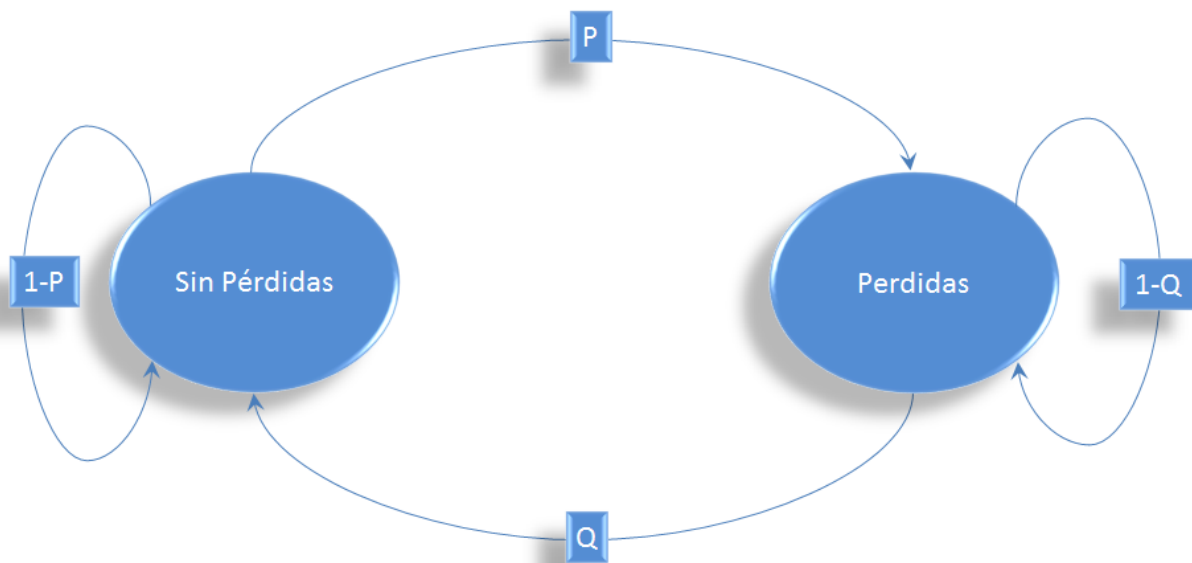


Figura 5.1: Modelo Gilbert-Elliot

Para cada uno de los vídeos originales se repetirá el proceso 10 veces para producir finalmente un total de 1380 vídeos degradados. Para poder llevar a cabo esta simulación se utilizará la aplicación Netem [24] con un parche para el sistema operativo para poder implementar el modelo de Gilbert-Elliot.

La primera vez que se realizó el experimento se retransmitió cada uno de los 23 vídeos con la aplicación VLC se detectó que los vídeos no se generaron correctamente, ya que VLC no mantiene la cantidad de paquetes necesarios para transmitir un vídeo de forma constante. Por lo tanto se recurrió a retransmitir por la interfaz *loopback* cada uno de los vídeos y retransmitirlos con el comando *tcpreplay* para cada una de las distintas tasas de pérdidas.

Una vez que se tiene el conjunto de vídeos simulados se pasa a calcular las pérdidas reales de cada uno de los vídeos, ya que aunque la red esté configurada para rechazar un tanto por ciento de los paquetes, también tiene una componente aleatoria que consigue tener una base de datos con una mayor variabilidad en su tasa de pérdidas, comprendiendo desde el 0% hasta un 12,5% de PLR.

Acto seguido, para estimar la calidad de los vídeos simulados existen dos posibilidades: una evaluación subjetiva de los vídeos con un panel de usuarios, cuyo coste en tiempo y económico es elevado, o una evaluación objetiva de los vídeos con un algoritmo preciso.

La solución escogida finalmente fue la segunda, ya que el tiempo requerido para la preparación del evento sería muy alto. Por lo tanto, como la solución escogida fue un análisis objetivo se debe elegir un algoritmo para tomar como referencia. Los algoritmos FR, que usualmente sirven en el desarrollo de los modelos sin referencia o con referencia reducida, serán los seleccionados para tener una guía de referencia a seguir. El algoritmo seleccionado en este experimento fue VQM por su buen comportamiento en vídeos MPEG.

La figura 5.2 muestra los resultados obtenidos entre una evaluación subjetiva y los resultados del modelo VQM. El coeficiente de correlación lineal de Pearson entre la calidad subjetiva y la calidad obtenida mediante el modelo VQM de las secuencias de vídeo es de 0.948.

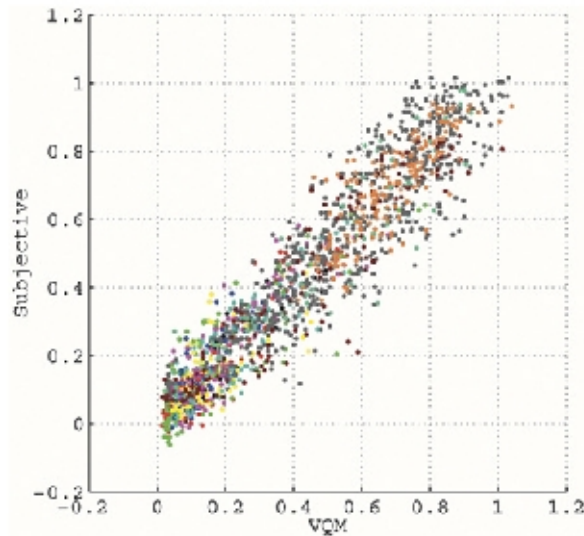


Figura 5.2: Calidad subjetiva vs modelo VQM [1]

Después de haber valorado cada uno de los vídeos con el algoritmo VQM se convirtieron los resultados obtenidos a una escala VQM. Para ello se ha seguido la fórmula 5.1.

$$MOS = 5 - 4 \cdot VQM \quad (5.1)$$

Una vez acabado esto, para terminar la base de datos sólo falta extraer las estadísticas generadas por la aplicación, tanto las estadísticas sobre la capa RTP como sobre la capa MPEG.

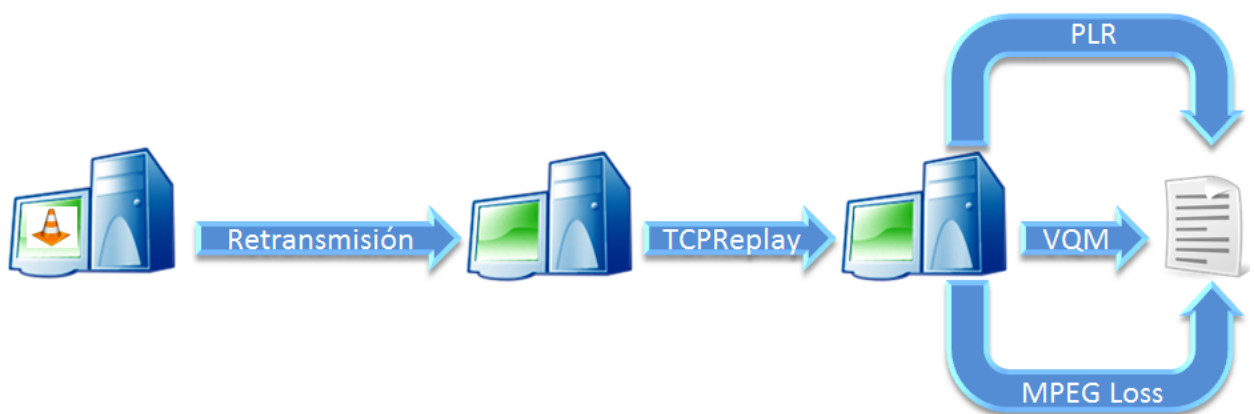


Figura 5.3: Modelo de creación de la base de datos

Teniendo finalizada la base de datos con todos los campos que queremos utilizar se puede pasar a calcular las fórmulas, así como su ajuste al modelo y su error.

5.3 Ajuste del modelo

En este apartado se estudiarán los modelos proporcionados y se calcularán los coeficientes deseados para dar por finalizada la aplicación. Primero se comentará la aproximación basada exclusivamente en la tasa de pérdidas. Después se dará una pequeña introducción a la codificación de cuadros en MPEG. Finalmente se mostrarán, los resultados obtenidos.

5.3.1 Ajuste sobre PLR

En este apartado se mostrará el algoritmo basado exclusivamente en las pérdidas de los paquetes, utilizado en la función `mos_model_plr`. Para ello se mostrarán en la Figura 5.4 los resultados extraídos del experimento especificado en el anterior apartado.

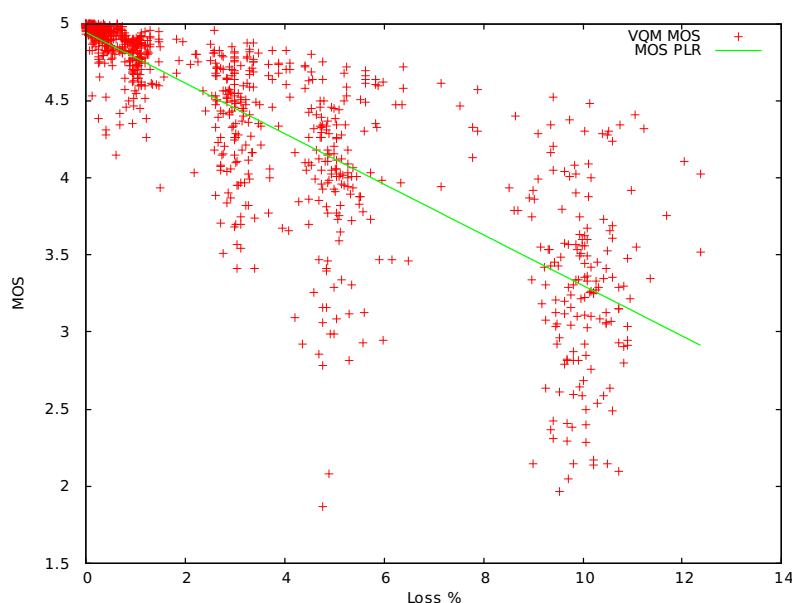


Figura 5.4: Modelo VQM MOS vs Modelo PLR

Como se puede apreciar en la figura, la tendencia de la calidad no es clara, pero se puede observar una pérdida de calidad en términos generales cuando las pérdidas de paquetes incrementan. Usando una regresión lineal, exponencial y logarítmica se intentó aproximar el valor del MOS. La función, de entre las barajadas, que mejor se ajusta al modelo fue la expresada en la función 4.1. Los coeficientes con un mayor ajuste a la función de MOS, calculado con el modelo VQM, son mostrados en la tabla 5.1. Estos datos han sido extraídos de las aplicaciones *Smith's Statistical Package* [25] y *Gnuplot* [26].

Cuadro 5.1: Resultados de la regresión lineal

Variable	Coefficiente	Std. Error	t Value	p-value
a	4.9442	0.0165	299.4776	0
b	-0.1642	0.0034	49.6947	0
SEE	0.3757			
R^2	0.6974			

Este modelo tiene un error cuadrático medio de 0.1412, en comparación con el modelo de VQM, lo que implica un error absoluto del 0.3757 (un 9.38 % de la escala MOS). Estos resultados son bastante positivos, ya que debemos considerar que el algoritmo opera sin tener conocimiento alguno del vídeo original. Por otro lado, se puede apreciar en la figura 5.1 que la función, basada en las pérdidas en la capa RTP, es una progresión de la media de los valores representados y no consigue adaptarse al comportamiento del MOS calculado.

5.3.2 Tipos de cuadros MPEG

Para obtener un mejor modelo del comportamiento del MOS, se indagó en la codificación de MPEG y se detectaron tres distintos tipos de cuadros:

- **I-frame:** Este tipo de cuadro es un fotograma convencional, una imagen en la que se incluye la información de toda la imagen que se debe de mostrar. Los otros tipos de cuadros necesitan la información incluida en las de tipo I. También son las que menor compresión y ratio de ocurrencia tienen. En la figura 5.5 se puede ver en el primer fotograma y el cuarto, cómo se repinta la pantalla completa.
- **P-frame:** Este es un tipo de cuadro predictivo, solo contiene los cambios producidos desde el anterior cuadro de tipo I o de tipo P. Por ejemplo, se puede ver la figura 5.5 cómo el cuadro de tipo P actualiza la posición de las bolas azules dejando intacto el resto del fotograma. Este tipo de cuadros tienen una ocurrencia y compresión mayor que las de tipo I e inferior a las de tipo B.
- **B-frame:** Este tipo de cuadros son predictivos pero de forma bidireccional, esto quiere decir que se necesitan información del cuadro previo y el posterior ya sea de tipo I o P. En la figura 5.5 se puede apreciar que para repintar la imagen requiere la información de la posición de las bolas del cuadro P, pero necesita la información de la trama I para incorporar una nueva bola.

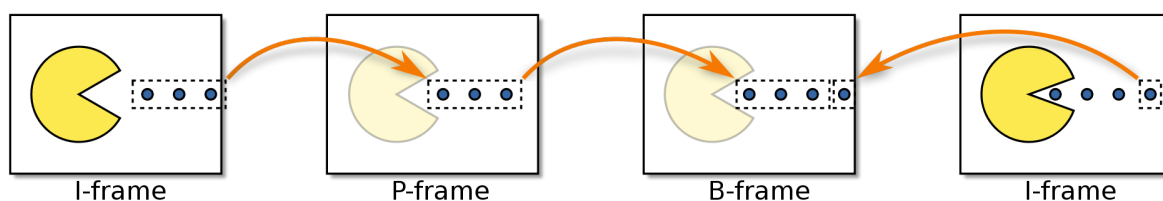


Figura 5.5: Uso de cuadros IBP

Como se puede imaginar, la pérdida en los distintos tipos de cuadros no tendrá el mismo peso y dependerá de la cantidad de aparición de estas, una trama de tipo B aparece con mucha más frecuencia, como de la cantidad de información transportada, la pérdida de una trama de tipo I influye en una gran cantidad de cuadros frente. A su vez, la distribución de pérdidas en estos tres tipos de cuadros no será equitativa a causa de la distribución aleatoria de las pérdidas generadas.

5.3.3 Ajuste sobre IBP

Una vez vistos los distintos tipos de cuadros existentes y la diferencia existente entre ellos, se obtienen los porcentajes de pérdidas para cada uno de los tipos, a través del analizador de tráfico desarrollado en 4.2.3.

Una vez realizado un análisis sobre la base de datos, se descubrió que los vídeos con un mayor porcentaje de pérdidas de tipo I eran castigados con una menor calidad. Esto es bastante lógico, ya que si se pierde este cuadro toda referencia a él también fallará y no se podrá apreciar bien la imagen.

Además, una pérdida en un cuadro de tipo P afectará a una cantidad de cuadros de tipo B y P. Por otra parte, una pérdida en un cuadro de tipo B no afecta a ningún otro cuadro. También cabe destacar que en los experimentos realizados la proporción de tipos de tramas es de un cuadro de tipo I por estructura GOP, 7 cuadros de tipo P y 14 cuadros de tipo B. Esto también puede afectar al peso asignado a las pérdidas de cada uno de los tipos.

Basados en esta idea, se desarrolla un modelo inicial, en el que las pérdidas porcentuales de los cuadros dependen de la cantidad de paquetes de la transmisión. Este modelo no funciona demasiado bien y se diseña uno nuevo.

Este nuevo modelo tiene en cuenta las pérdidas de cada tipo de cuadro en función al número total de cuadros leídos de cada tipo. Este modelo comienza a dar unos resultados mejores, se pueden observar en la figura 5.6 (los valores representados, tanto en esta figura como en la figura 5.8 están expresados en tanto por 1). La tabla 5.2 muestra los coeficientes encontrados para la regresión lineal, después de realizar una nueva comparativa entre una regresión lineal, exponencial y logarítmica.

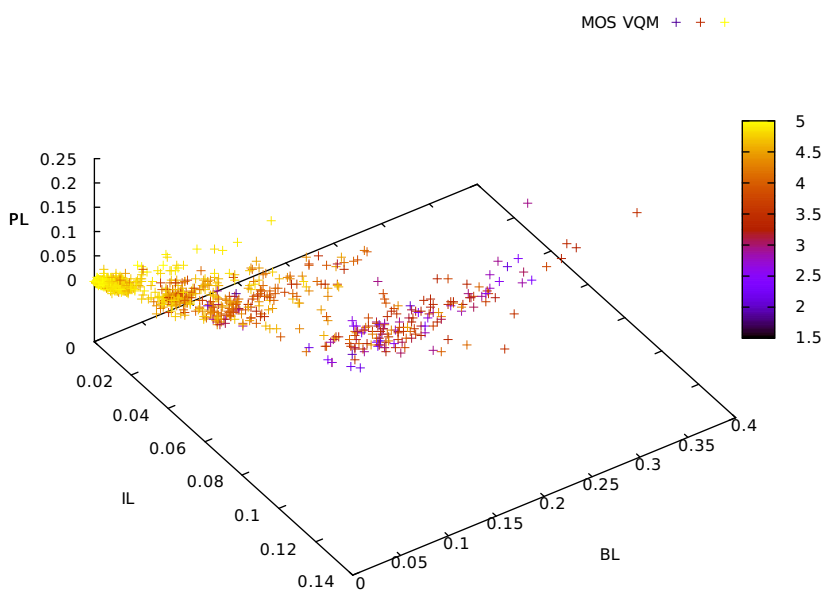


Figura 5.6: Primer modelo IBP

Cuadro 5.2: Coeficientes y resultados de la regresión lineal IBP (1)

Variable	Coefficiente	Std. Error	t Value	p-value
<i>a</i>	4.9437	0.0165	298.7312	0
<i>b</i>	-17.5488	1.5542	11.2913	0
<i>c</i>	-0.8657	0.3576	2.4212	0.0078
<i>d</i>	1.6701	0.9601	1.6738	0.0472
SEE	0.3764			
R^2	0.6969			

El nuevo modelo presentado tiene un comportamiento más parecido al del MOS calculado con el modelo VQM, pero los resultados recogidos son peores en comparación con el modelo de IBP con un error cuadrático medio de 0.1416, lo que implica una un error medio de 0.3764 (9.5% de la escala MOS). Además el valor de R^2 tiene un valor inferior del modelo PLR lo que implica que este nuevo modelo tiene una peor correlación que el modelo PLR. En la figura 5.7 se puede ver la representación del primer modelo de IBP frente al modelo PLR.

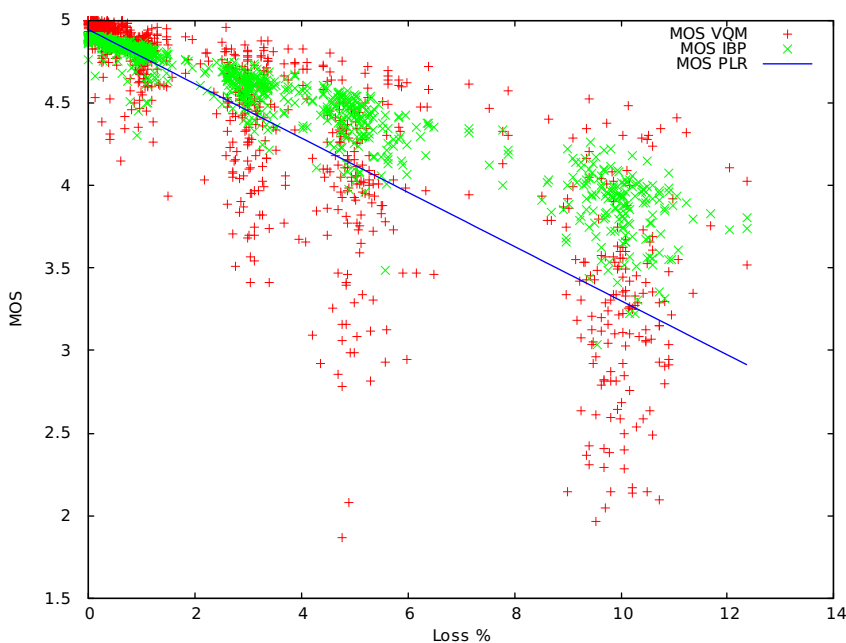


Figura 5.7: Primer modelo IBP

Para mejorar el modelo obtenido, se investigó sobre el funcionamiento de los codificadores de vídeo actuales y se descubrió que la mayoría de ellos desechan el resto de la información del cuadro si se pierde la cabecera. Sin embargo si se pierde información interna el codificador puede seguir funcionando correctamente. Por lo tanto se modificó el código para poder capturar las pérdidas según el nuevo modelo, como se puede observar en la figura 5.8.

En esta figura se puede ver un comportamiento de la calidad separado por regiones, un cono que tiene su punta en el punto de no pérdidas y que se va expandiendo según aumentan las pérdidas de cada tipo, permitiendo la diferenciación de los distintos grupos de valores de calidad de experiencia.

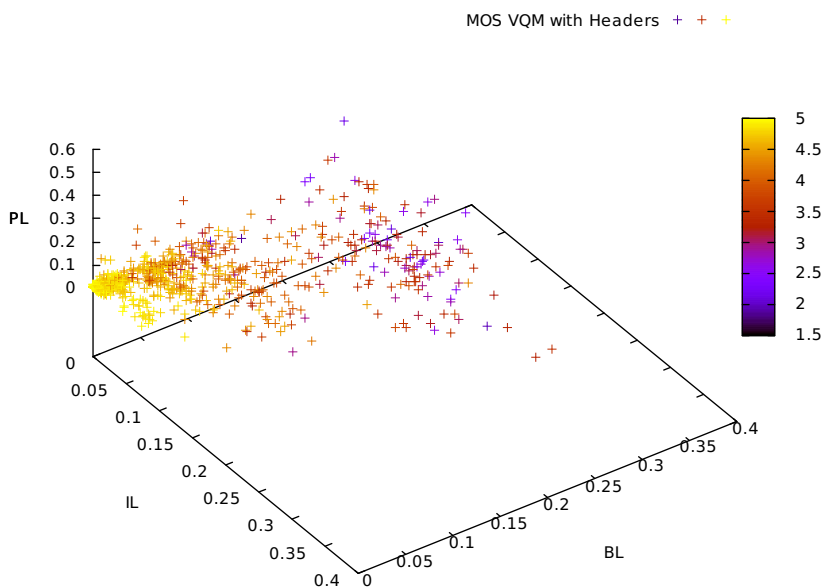


Figura 5.8: Segundo modelo IBP

De nuevo, se intenta ajustar el modelo VQM con el nuevo grupo de datos a través de tres regresiones: lineal, exponencial y algorítmica. El resultado vuelve a ser que el modelo lineal es el que mejor se ajusta a los datos obtenidos. En la tabla 5.3 se pueden ver los coeficientes seleccionados para la función 4.2.

Cuadro 5.3: Coeficientes y resultados de la regresión lineal IBP (2)

Variable	Coefficiente	Stdr. Error	t Value	p-value
a	4.9030	0.0142	344.9143	0
b	-1.0823	0.2935	3.6876	0.0001
c	-3.2792	0.4759	6.8911	0
d	-3.2323	0.2784	11.6111	0
SEE	0.3367			
R^2	0.7575			

Este nuevo modelo tiene un valor de error cuadrático medio de 0.113 con respecto al modelo VQM, con un error absoluto de 0.3367 (un 8.4% en la escala MOS). Además, el valor de R^2 es de 0.7575.

Estos valores confirman que este modelo tiene un mejor comportamiento frente al modelo VQM que el modelo PLR y el anterior modelo IBP. Tiene un menor error cuadrático, casi un 1 % menor que el modelo PLR, pero lo más importante es el incremento del valor de R^2 , lo que implica que esta función explica mejor el comportamiento del modelo VQM, como se puede ver en la figura 5.9).

Cabe destacar que la diferencia de estos resultados con los anteriormente obtenidos puede estar relacionado con dos factores principalmente. El primero, que el tener en cuenta las pérdidas de las cabeceras conlleva a un modelo más realista. La segunda, es la relación variable entre el tamaño de las cabeceras y el cuadro entero.

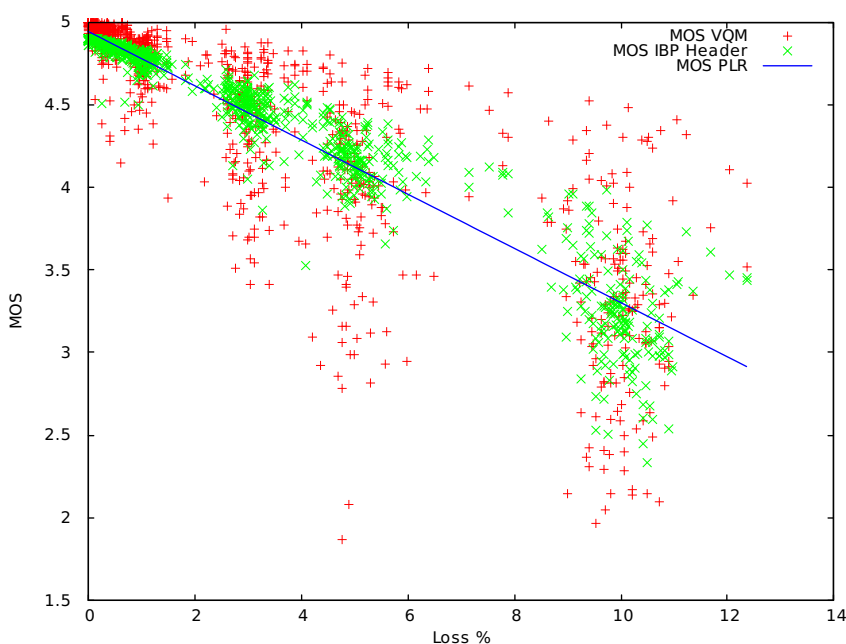


Figura 5.9: Segundo modelo IBP

Además, cabe analizar que el valor de los coeficientes obtenidos puede ser algo diferentes a los esperados (un peso muy alto en cuadros de tipo I y un menor peso en los de tipo B). Esto puede deberse a la cantidad de tramas de tipo B y P así como su gran tamaño en los experimentos realizados. En experimentos posteriores se deberá incluir vídeo de alta definición para comprobar si estos valores se mantienen.

5.4 Rendimiento

La última cuestión que queda por resolver es si la mejora de estimación obtenida del nuevo modelo compensa la sobrecarga que añade la extracción de estadísticas de la capa MPEG. Esta extracción de estadísticas añade a la ejecución la detección de los tipos de cuadros MPEG y una máquina de estados para la detección de estas.

Esta sobrecarga en el código se puede apreciar en la figura 5.10 en la que se ha realizado un análisis del tiempo que necesita cada uno de los dos modelos, para extraer las estadísticas y calcular la calidad deseada, en función del número de capturas, de 10 segundos de duración, analizadas. En este análisis se han utilizado las capturas extraídas de la integración de ALU (véase capítulo 6).

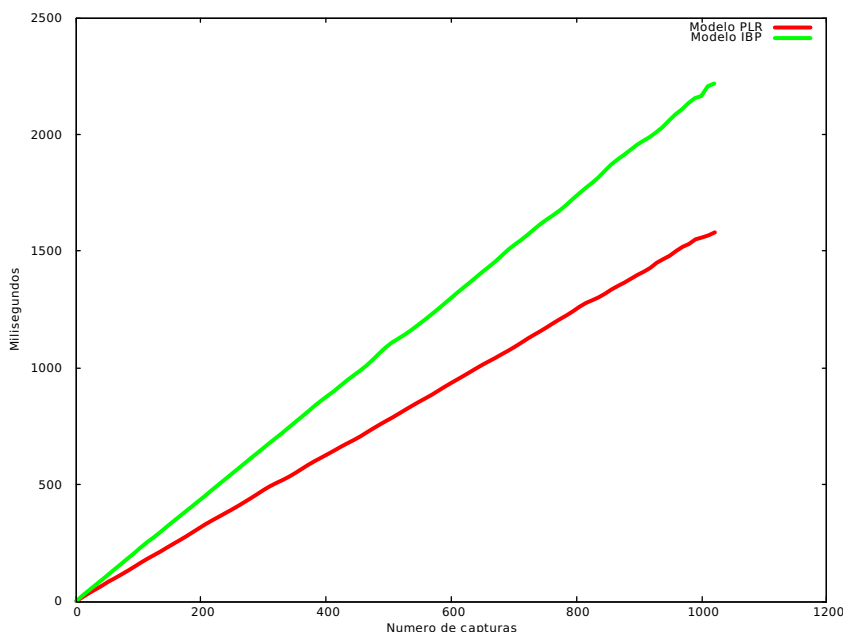


Figura 5.10: Prueba de rendimiento

Los datos obtenidos son muy positivos. Aunque el modelo PLR es más eficiente que el modelo basado en MPEG, el nuevo modelo es capaz de analizar más de 300 capturas, de 10 segundos de duración, en menos de 1 segundo. Aun así, hay que tener en cuenta que los datos han sido extraídos de un ordenador con 4GB de memoria RAM y 8 núcleos a 2.8GHz. La aplicación solamente utiliza un *core*, pero la memoria RAM sí que influye de una manera importante, ya que las capturas de tráfico son almacenadas en RAM, no en disco directamente, lo que aumenta ampliamente la velocidad de escritura y lectura de estos ficheros.

Posteriormente, se realizó un segundo *test* en el que se utilizó un vídeo en HD (1920 x 1080) [27]. Este vídeo repitió el proceso de la creación de la base de datos (codificación MPEG, simulación de pérdidas, y troceado en fragmentos de 10 segundos). Este vídeo se utilizara para comprobar la velocidad con canales HD, ya que la resolución de los anteriores canales era inferior. En la figura 5.11 se pueden apreciar los resultados.

Como se puede ver en la figura 5.11 el tiempo que requiere del modelo IBP sigue siendo superior al modelo más simple, pero es capaz de analizar 100 capturas de 10 segundos de vídeo en menos de 2 segundos. Esta aplicación se puede seguir llevando a un sistema de bajas prestaciones si no se quiere analizar más de 100 canales en HD, que es un número de canales bastante elevado. Aún cabría la posibilidad de paralelizar el análisis de capturas para aumentar la eficiencia.

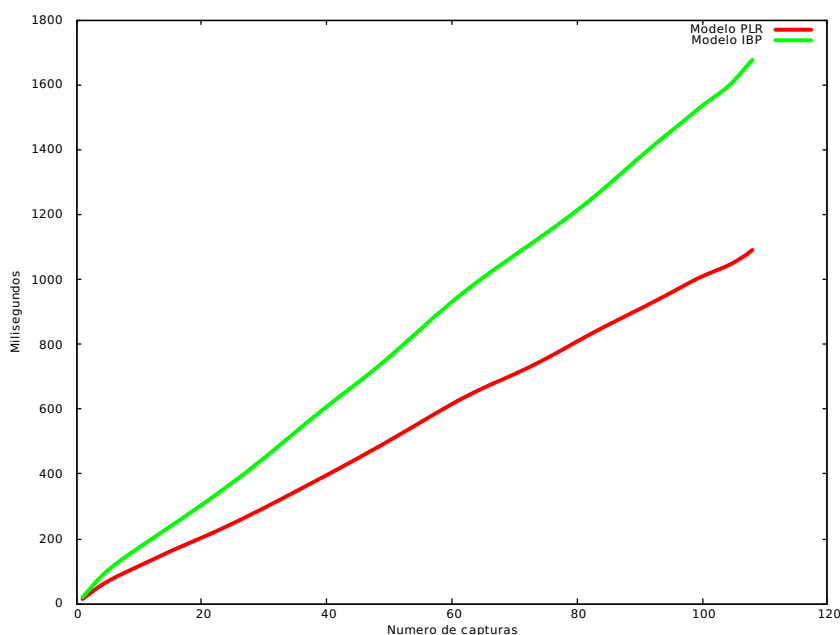


Figura 5.11: Prueba de rendimiento HD

Si finalmente la aplicación se llevara a una sonda, se debería tener en cuenta que las características de la máquina no serían iguales. La sonda deberá ser un aparato de bajo coste, con características similares a las siguientes:

- Uno o dos cores (uno para el capturador y otro para el analizador de tráfico) a una velocidad de unos 700MHz. Esto implicaría una velocidad de procesamiento 4 veces inferior, aun así la aplicación podría ser capaz de analizar teóricamente alrededor de 90 capturas, de 10 segundos, en un segundo.
- Una memoria RAM de entre 1GB y 2GB, ya que se proveería de un gestor de memoria que volcase el contenido de la memoria RAM a disco, evitando el *swap* de memoria RAM en disco. Además, una captura de 10 segundos de un canal en HD no debería ocupar más de 50MB, lo que da margen de actuación. Si se quieren analizar menos de 10 canales un tamaño de memoria RAM inferior a 1GB debería bastar.
- Una interfaz de red, a través de la que se recibe una copia del canal recibido en el televisor y se envían las estadísticas.

Como se ha mostrado, las características de la sonda no deben ser demasiado exigentes. Existen modelos parecidos de venta al público con precios muy reducidos, con costes similares a un *router* sencillo. Además cabe investigar la integración de esta aplicación a un *router* como el que proporciona el proveedor de servicios.

5.5 Conclusiones

En este apartado se ha podido ver una explicación algo más exhaustiva del contenido del artículo publicado (véase Anexo A).

Primero se ha mostrado el método seguido para la creación de la base de datos necesaria para las pruebas, tanto la degradación de vídeos como el método de cálculo del MOS. Después se ha creado un modelo basado en las pérdidas exclusivamente (válido para cuando los canales no estén codificado en un formato soportado). Además, se ha tratado de aproximar un modelo a través de las pérdidas en los distintos tipos de cuadros de la codificación MPEG y se ha realizado una comparativa entre los modelos.

Finalmente, se han comparado la eficiencia de los dos modelos de estimación de calidad. Como se puede observar, el modelo más simple tiene un mejor comportamiento, pero el segundo modelo es capaz de procesar 1000 capturas en menos de 3 segundos o 100 capturas en HD en menos de dos segundos. Estos resultados indican que se podría crear en un trabajo futuro un modelo computacionalmente más complejo, ya que el cuello de botella de la aplicación no se encuentra ahí, sino en el ancho de banda (para el análisis de varios canales).

En el siguiente capítulo se mostrará la integración del código en un entorno más realista, donde se introducirá la sonda en una arquitectura donde se simulará un modelo más realista.

6

Integración

6.1 Introducción

En este capítulo se relatará la integración del código desarrollado dentro del grupo de empresas pertenecientes al consorcio **PRINCE**. Se mostrará la arquitectura diseñada para la simulación, la configuración necesaria para poder ejecutar satisfactoriamente la aplicación y se mostrarán los errores que se han ido detectando y solucionando.

6.2 Proyecto PRINCE

El trabajo realizado en este TFG ha sido llevado a cabo dentro del consorcio **PRINCE** (PProducto INdustrial para la gestión de Calidad de Experiencia), perteneciente a la iniciativa internacional **IPNQSIS** (IP Network Monitoring for QoS Intelligent Support) [28]. Esta iniciativa fue promovida principalmente por de Celtic, EUREKA y parcialmente financiada por el CDTI [29]. El proyecto comenzó a plantearse en el año 2009 y finaliza este año.

IPNQSIS cuenta con la intervención de un grupo de empresas españolas, entre las que se encuentran Indra, Alcatel-Lucent, Naudit, Dycec, Broadcom y SoftTelecom. Además cuenta con empresas de otros tres países europeos: Suecia, Francia y Finlandia.

El proyecto se ha centrado principalmente en la monitorización y la gestión de los servicios ofertados en la llamada “Nueva generación de conexiones IP” (IPTV, televisión móvil, VoIP, etc.) desde la perspectiva de los proveedores, proporcionándoles un control de la calidad de experiencia y asegurándoles unos niveles óptimos de calidad.

En el sector en el que se ha trabajado, el trabajo se ha centrado en la predicción de la calidad de experiencia del usuario final a través de algoritmos NR. El grupo con el que se ha trabajado estaba formado por Alcatel-Lucent, Broadcom, Dycec, Naudit y SoftTelecom. La UAM ha participado en este proyecto subcontratada por la empresa Naudit.

A continuación se mostrará el diseño seguido para la integración y prueba de los distintos modelos creados por los participantes del proyecto.

6.3 Arquitectura

Para probar las soluciones creadas por cada uno de los grupos participantes, se diseña una arquitectura dónde se puede simular el correcto funcionamiento de cada una de las sondas. Como se puede ver en la figura 6.1, se dispone de un generador de vídeo con un módulo de simulación de pérdidas configurable y una selección de distintos tipos de canales (deportes, informativos, dibujos).

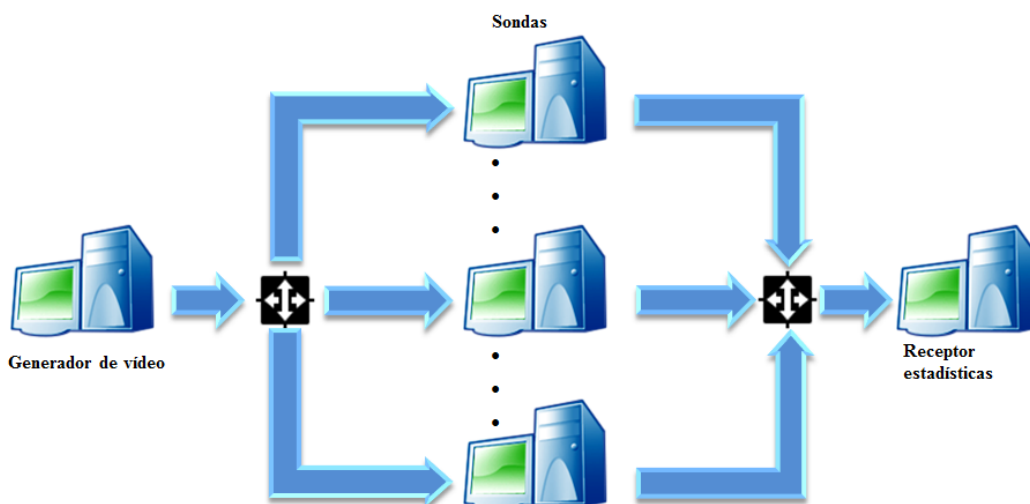


Figura 6.1: Arquitectura de la simulación

Esta variedad de canales es una manera de probar el sistema de una manera bastante completa, ya que la cantidad de movimiento varía mucho de un canal a otro y esto aumenta o disminuye la influencia de una pérdida en la calidad. Posteriormente, esta información se distribuye a través de un *switch* a cada una de las sondas para que analicen el contenido del canal. Cada sonda analiza el contenido del canal con su propio modelo y envía la información al receptor de estadísticas a través del servicio web GSOAP.

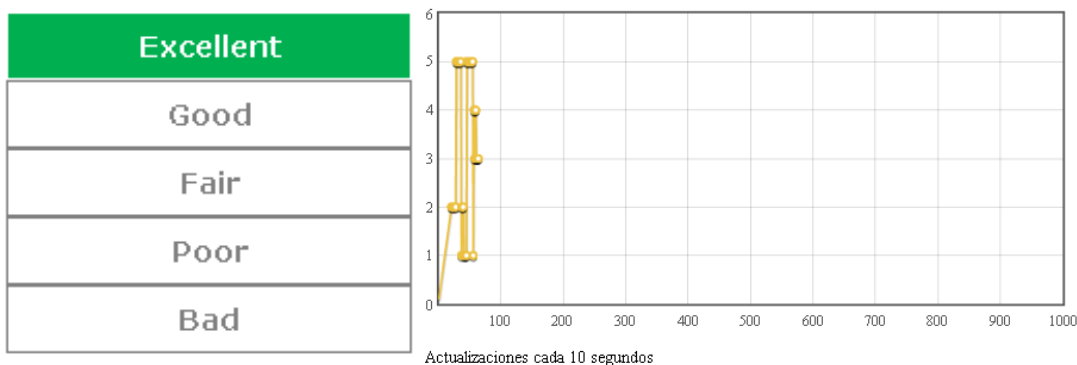


Figura 6.2: Pagina Web

Cada vez que transcurre el intervalo de tiempo predeterminado, cada una de las sondas envía la información al receptor de estadísticas a través del servicio web. Esta comunicación se realiza a través del protocolo de comunicación SOAP. Este protocolo establece un estándar de cómo dos aplicaciones pueden comunicarse a través de una sistema de ficheros XML. Las estructuras y las funciones necesarias para la comunicación se describen en un fichero WSDL que deben tener los dos extremos de la comunicación.

Finalmente, cada vez que transcurre un intervalo de tiempo prefijado se realiza una media de los cálculos de las sondas y se muestra a través de una página web los resultados obtenidos, como se puede ver en la figura 6.2. La página consta de dos partes: la primera es un semáforo donde se indica cual es el MOS actual, la segunda es una gráfica donde se va mostrando la calidad obtenida a lo largo del tiempo.

6.4 Sonda

La sonda que ejecuta la aplicación desarrollada, que se ha colocado dentro de la arquitectura mostrada en la figura 6.1, es un ordenador de uso común con tres interfaces de red distintas: dos interfaces cableadas y una interfaz inalámbrica.

La primera interfaz cableada es utilizada para la captura de datos de vídeo. La segunda interfaz se utiliza para la comunicación con el servidor de estadísticas. Finalmente, la interfaz inalámbrica es utilizada para la comunicación con el exterior de la sonda, para modificaciones de código o análisis de fallos.

En la sonda se están ejecutando de forma indefinida el módulo capturador de canales, el analizador de tráfico y el módulo encargado de la comunicación y de la estimación de la calidad, explicados en el capítulo 4. Paralelamente, se está ejecutando un recolector de basura, para poder liberar de espacio al disco (ocupado por capturas erróneas almacenadas), si el espacio ocupado sobrepasa unos límites preestablecidos.

Finalmente, para habilitar la comunicación con el exterior se está ejecutando una aplicación que comprueba y reinicia, si fuera necesario, dos túneles SSH inversos. Este túnel es necesario, ya que la sonda se encuentra en una red interna a la que no se puede acceder directamente desde el exterior. El segundo túnel funciona de comodín en caso de que el primer túnel pierda la conexión.

6.5 Conclusiones

En este capítulo se ha relatado la integración de la aplicación desarrollada en un entorno simulado con otros modelos de estimación de calidad. Se ha explicado la arquitectura diseñada para la simulación y el funcionamiento interno de la sonda. El último paso que faltaría dar sería llevar la sonda a una simulación más compleja y comprobar su funcionamiento.

Por último, se mostrará en el último capítulo del documento las conclusiones extraídas del trabajo realizado y se mostrarás las tareas que aún quedan por hacer para perfeccionar la aplicación.

7

Conclusiones y Trabajo futuro

7.1 Introducción

En este capítulo se analizará el trabajo desarrollado y expuesto en este documento y se extraerán las conclusiones finales del documento. Además, se analizarán los puntos que necesitan un trabajo extra (mejoras visuales, integración, etc.) en la sección de Trabajo Futuro.

7.2 Conclusiones

El objetivo de este TFG era la creación de un modelo de estimación de calidad que cumpliera una serie de características (eficiencia, sin consumo de red, etc.). El resultado obtenido ha sido una aplicación fácilmente configurable, que consigue unos resultados aceptables a bajo coste.

La aplicación se ha dividido en tres módulos fácilmente configurables: receptor de flujos, analizador de tráfico y el módulo encargado de la comunicación y el cálculo de la calidad. Esta división ha hecho mucho más sencillo tanto el desarrollo del código como la prueba de su funcionamiento y su integración.

El módulo de cálculo de calidad es capaz de generar estimaciones de calidad de experiencia del usuario final con un porcentaje de error reducido y tiene un comportamiento similar a la calidad calculada con el algoritmo VQM. Para la estimación de esta calidad se ha optado por la creación de dos modelos diferentes. El primero se ha basado su estimación en el ratio de pérdidas de paquete en una ventana de tiempo predeterminada. El segundo modelo, se basa para calcular la estimación en las pérdidas relativas de los distintos tipos de cuadros existentes en MPEG.

Además, se ha comprobado que el módulo analizador de tráfico y el módulo estimador de calidad en conjunto son capaces de procesar una gran cantidad de flujos de vídeo simultáneamente, ya sea de calidad media o HD. Todo esto implica que no se necesita un supercomputador para poder procesar estos canales de vídeo, ya que como se ha podido ver en la sección 5.4 las características estimadas de la sonda propuesta son muy similares a las de dispositivos que se pueden encontrar en el mercado por un reducido precio.

Posteriormente, la aplicación se ha integrado en un entorno simulado, donde se pudo comparar con otras soluciones desarrolladas, y se probó su correcto funcionamiento. En este punto se ha podido comprobar la cantidad de problemas que surgen a lo largo de una integración. Uno de los principales problemas fue conseguir habilitar la conexión con el servidor de estadísticas Java, ya que como se ha dicho previamente, los tipos de datos sin signo utilizados tuvieron que ser sustituidos por tipos de datos de mayor tamaño.

Además, se añade el problema de tener el acceso a la sonda principalmente a través de la red. Este problema implica la reparación de módulos de forma local, una posterior integración en el sistema real y una dependencia total de la conexión a Internet de la sonda.

Finalmente, se ha realizado una aplicación con unos resultados aceptables para las condiciones de funcionamiento finales que podría tener y se ha extraído una experiencia muy positiva de la realización de este TFG. Se han aprendido nuevas tecnologías, se han cometido errores y también se ha aprendido a afrontarlos. Aún así, ningún trabajo es perfecto y se podrían haber realizado una serie de mejoras y/o estudios que habrían hecho el trabajo más completo, todas las que se han ido estudiando a lo largo de este TFG se plantean en la siguiente sección.

7.3 Trabajo Futuro

Para mejorar los puntos más débiles de este documento, queda abierta la realización de una amplia gama de trabajos futuros. Entre ellos se encuentran:

- Crear una base de datos con mayor diversidad de vídeos. Debe haber una gran gama de vídeos con diferentes calidades, distinto movimiento, distinto tipo de contenido (dibujos, deportes, informativos, etc.) y pérdidas. Además, se deberán crear dos bases de datos, una para el entrenamiento del modelo y otra para la prueba del modelo creado.
- Crear un panel de usuarios que evalúen los vídeos obtenidos para poder tener unos datos de calidad altamente fiables. Si esto no se puede conseguir, se debería crear una base de datos que contuviese el resultado del análisis de los vídeos por varios algoritmos.
- Investigar la influencia que tiene la cantidad de movimiento de un vídeo en el peso de las pérdidas del canal. No es lo mismo una pérdida en un canal con mucha cantidad de movimiento (deportes) que una pérdida en un canal con poco movimiento (informativo).
- Investigar la paralelización del algoritmo de análisis del vídeo para aumentar la cantidad de canales que se puede analizar por segundo y comprobar la mejora de rendimiento.
- Investigar el uso de modelos matemáticos más complejos para la estimación de la calidad, ya que el algoritmo permite invertir una mayor cantidad de tiempo en el cálculo de la calidad, dada la alta velocidad de procesamiento de canales.
- Estudiar a fondo las características que debe tener la sonda final para poder llevar la aplicación a los puntos finales de la red. También cabría estudiar la capacidad de procesamiento y memoria de los routers que actualmente dispone un usuario medio en su hogar y ver si se puede integrar la aplicación en ellos.
- Mejorar la interfaz web de recepción de estadísticas y el servicio web, con una gestión de usuarios, consultas de calidades antiguas, capacidad de descargar una versión comprimida del vídeo, etc. (La maqueta realizada para la interfaz web puede verse en el Anexo B).

Bibliografía

- [1] M. Pinson and S. Wolf, “A new standardized method for objectively measuring video quality,” *Broadcasting, IEEE Transactions on*, vol. 50, no. 3, pp. 312–322, 2004.
- [2] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz, “Quality of service terminology in ip networks,” *Communications Magazine, IEEE*, vol. 41, no. 3, pp. 153–159, 2003.
- [3] A. Van Moorsel, “Metrics for the internet age: Quality of experience and quality of business,” in *Fifth International Workshop on Performability Modeling of Computer and Communication Systems*, vol. 34, no. 13, Nürnberg, Germany, 2001, pp. 26–31.
- [4] “Quality of experience for mobile video users.” [Online]. Available: <http://www2.tkn.tu-berlin.de/research/evalvid/qcif.html>
- [5] U. Engelke and H.-J. Zepernick, “Perceptual-based quality metrics for image and video services: A survey,” in *Next Generation Internet Networks, 3rd EuroNGI Conference on*, 2007, pp. 190–197.
- [6] “Ntt.” [Online]. Available: <http://www.nttmcl.com/sec/>
- [7] “Opticom.” [Online]. Available: <http://www.opticom.de/>
- [8] “Psytechnics.” [Online]. Available: <http://www.psytechnics.com/>
- [9] “Yonsei.” [Online]. Available: <http://www.yonsei.ac.kr/eng/>
- [10] K. Seshadrinathan and A. Bovik, “Motion tuned spatio-temporal quality assessment of natural videos,” *Image Processing, IEEE Transactions on*, vol. 19, no. 2, pp. 335–350, 2010.
- [11] K. Seshadrinathan, R. Soundararajan, A. Bovik, and L. Cormack, “Study of subjective and objective quality assessment of video,” *Image Processing, IEEE Transactions on*, vol. 19, no. 6, pp. 1427–1441, 2010.
- [12] M. Venkataraman and M. Chatterjee, “Inferring video qoe in real time,” *Network, IEEE*, vol. 25, no. 1, pp. 4–13, January-February.
- [13] F. Yang and S. Wan, “Bitstream-based quality assessment for networked video: a review,” *Communications Magazine, IEEE*, vol. 50, no. 11, pp. 203–209, 2012.
- [14] P. de la Cruz Ramos, J. Navarro Salmerón, R. Pérez Leal, and F. González Vidal, “Estimating perceived video quality from objective parameters in video over ip services,” April 29, 2012, pp. 65–68.
- [15] J. Pokhrel, B. Wehbi, A. Morais, A. Cavalli, and E. Allilaire, “Estimation of qoe of video traffic using a fuzzy expert system,” in *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, 2013, pp. 224–229.

- [16] M. Fiedler, T. Hossfeld, and P. Tran-Gia, “A generic quantitative relationship between quality of experience and quality of service,” *Network, IEEE*, vol. 24, no. 2, pp. 36–41, 2010.
- [17] “Ipfix recomendación.” [Online]. Available: <http://www.iana.org/assignments/ipfix/ipfix.xml>
- [18] “Ietf recomendación.” [Online]. Available: <http://tools.ietf.org/html/draft-scholz-ipfix-rtp-msg-00>
- [19] “Yuv qcif reference videos.” [Online]. Available: <http://www2.tkn.tu-berlin.de/research/evalvid/qcif.html>
- [20] “Ffmpeg.” [Online]. Available: <http://www.ffmpeg.org/>
- [21] G. Hasslinger and O. Hohlfeld, “The gilbert-elliott model for packet loss in real time services on the internet,” in *Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference -*, 31 2008-April 2, pp. 1–15.
- [22] M. Ellis and C. Perkins, “Packet loss characteristics of iptv-like traffic on residential links,” in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, Jan., pp. 1–5.
- [23] M. Borella, D. Swider, S. Uludag, and G. Brewster, “Internet packet loss: measurement and implications for end-to-end qos,” in *Architectural and OS Support for Multimedia Applications/Flexible Communication Systems/Wireless Networks and Mobile Computing., 1998 Proceedings of the 1998 ICPP Workshops on*, Aug, pp. 3–12.
- [24] “Netem.” [Online]. Available: <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>
- [25] “Smith’s statistical package.” [Online]. Available: <http://economics-files.pomona.edu/StatSite/SSP.html>
- [26] “Gnuplot.” [Online]. Available: <http://www.gnuplot.info/>
- [27] “Big buck bunny.” [Online]. Available: <http://www.bigbuckbunny.org/>
- [28] “Ipnqsis project.” [Online]. Available: http://ipnqsis.org/index.php?title=Main_Page
- [29] “Cdti.” [Online]. Available: <https://www.cdti.es/>



Artículo publicado

En este anexo se adjunta el artículo que se publicará en la 4ª conferencia de **SaCoNet** (Smart Communications in Network Technologies), específicamente en el primer seminario internacional de monitorización de calidad (IWQM). Este seminario se llevará a cabo el 17 de Junio de 2013 en París y está avalado por el comité técnico de SaCoNet 2013 y EU Eureka Celtic IPNQSIS.

El objetivo principal de este seminario será el de presentar los resultados de la investigación y el estado del arte en el contexto de la monitorización y gestión de la calidad de experiencia. IWQM 2013 combinará los resultados de los trabajos sobre las iniciativas europeas enfocadas en este ámbito. Además, los asistentes a las conferencia estarán estimulados para participar en discusiones abiertas en las que cada persona pueda compartir sus ideas.

Dentro de este seminario, el trabajo realizado se enmarcará en el grupo de trabajos que se encargan del análisis y monitorización de calidad de experiencia. En concreto, en el grupo de análisis de la red.

La primera versión del artículo se envió el día 30 de marzo de 2013. Posteriormente se obtuvieron los resultados de las revisiones efectuadas por el comité de evaluación, de los que se extraen un trabajo con gran relevancia dentro del seminario, un contenido técnico y rigor científico aceptable, una originalidad aceptable y una calidad de presentación buena.

Posteriormente, se corrigieron los defectos señalados por los revisores, en su gran mayoría referidos a aspectos visuales o frases que no quedan lo suficientemente claras, y se incluyeron, en la sección de trabajo futuro, las mejoras que no eran planteables en un corto espacio de tiempo. Finalmente queda como trabajo pendiente realizar la presentación el día 17 de Junio para terminar el trabajo relativo al artículo.

En la siguiente página se puede adjuntado el artículo referido en esta página. Por último, se adjunta la carta de aceptación del artículo.

Evaluating Quality of Experience in IPTV Services Using MPEG Frame Loss Rate

D. Hernando, J.E. López de Vergara, D. Madrigal, F. Mata
High Performance Computing and Networking Research Group,
Escuela Politécnica Superior, Universidad Autónoma de Madrid
Madrid, Spain
Email: diego.hernando@estudiante.uam.es,
{jorge.lopez_vergara, david.madrigal, felipe.mata} @uam.es

Abstract—Current network services such as Voice over IP or IP Television pose new challenges to network providers. Network operators need to know if their services are being properly provided. However, current quality of service parameters commonly used in data networks (e.g. throughput, packet delays, packet losses, etc.) do not show a clear view of what the users are experiencing. Thus, it is necessary to translate such measured quality parameters into a quality of experience value. Several models are being developed to cope with this problem. For instance, some approaches have used the packet loss rate to evaluate the experienced quality of an IP television channel. Unfortunately, packet loss just explains a fraction of the quality behavior. Then, we go one step further, taking into account the different MPEG frame types that are transmitted. In this paper, we have defined a model to predict the experienced quality that is a function of the loss of the different types of MPEG frames, providing a mean opinion score of the delivered service. The final results show that our model is able to better predict the quality of experience of such video services than just using the packet loss rate.

Index Terms—Quality of Experience, Video quality, VQM, packet loss, Mean Opinion Score, multiple lineal regression.

I. INTRODUCTION

Network services based on Internet technology are growing and evolving every day. For example, most network providers have in their portfolio services such as IP television (IPTV), Voice over IP (VoIP), etc. With millions of people using these technologies, network operators need a way to check such services are being properly delivered to the end users. This is important, because their business is based on the quality they can provide. Then, these operators need effective methods for measuring such quality. Current measurements are focused on network parameters that provide information about the quality of service (QoS) [1]. For instance, if a link is up and running, or the throughput that is achieved on such link. This information is useful to diagnose the current status of the network and reduce time to repair the infrastructure. Nevertheless, such QoS information does not map well to what the service users are finally perceiving.

To solve this problem, the quality of experience (QoE) [2] is a way to measure how well the services are being provided to the customers, based on users' opinion. However, the transformation from QoS parameters to QoE parameters is not direct. Several models have been defined in the literature related to map QoS into QoE. For instance, related to video

services, we can find the Video Quality Model (VQM) [3], which provides information about the experienced quality of the customers of a video service. The VQM has a good correlation to the QoE, but it has the problem that it consumes many resources to provide a measurement, making unviable to use it for assessing deployed network services.

A feasible QoE measurement must not consume many resources, such as CPU or bandwidth, letting it run on general purpose computers, which can be deployed along the network. To accomplish this purpose, we have investigated a method to translate QoS network measurements into QoE measurements. This method is based on the following steps: First of all, we have created a large database of video sequences with different packet loss rates. Then, we have measured their QoE comparing such video sequences with their original source, using the VQM. Next, we have searched for a relationship of the obtained QoE value with the packet loss rates, including also the influence of MPEG frame types in such loss rates. Finally, based on these results, we have been able to define a method to measure the QoE by capturing the live video channels, inspecting the packets to detect losses, and applying the measured parameters with the obtained model.

The rest of the paper is structured as follows: Next section is about related work, explaining the different alternatives to obtain a QoE model. Then, the method used to obtain our model is provided in section III. After this, section IV shows the obtained results. Finally, some conclusions are provided.

II. RELATED WORK

Traditionally, operators have focused on measuring certain quality of service parameters (packet loss, delay and delay variation, mainly), to ensure that their value ranges are between those defined in certain pre-agreed Service Level Agreements.

However, it has been found that certain values ensuring high quality of service do not guarantee a high quality user experience. Moreover, there is no clear relationship between these QoS parameters and the quality of experience perceived by the user, which is also dependent on the type of service which is being evaluated.

In the case of multimedia services, there are two main

methodologies for evaluating the quality of experience: subjective evaluation and objective evaluation.

Subjective evaluation involves performing QoE questionnaires to a representative sample of users, which assess test sequences that are representative of the provided service. For this evaluation, the Mean Opinion Score (MOS) has been defined. It is a subjective quality rating on a scale from 1 to 5, 1 meaning the worst perceived quality, and 5 the highest possible quality. Users in subjective tests assign to each observed sequence a score in this scale, and the final score of the testing sequence is the average of the scores given by the users. To obtain statistically significant results, it is necessary to have a large sample of users, making the procedure highly expensive.

The disadvantages of the subjective evaluation have promoted interest in improving objective methodologies. Depending on the available information of the algorithm, objective methodologies can be divided into three categories [4]:

- 1) **Full Reference (FR)** algorithms have access to the original video and the degraded sequence. They can compare pixel by pixel to measure the QoE. FR measurements have the highest accuracy.
- 2) **Reduced Reference (RR)** algorithms use a fraction of the link capacity to send some parameters to predict the QoE without the original video. RR measurement have less accuracy but they do not need the original video, but just some parameters of it.
- 3) **No Reference (NR)** algorithms use the network information to predict the QoE. The measurement is limited to a transport stream analysis, but it does not need any reference of the original video.

It is not feasible to use a FR model in practice, because it is necessary to have the original video. However, such models are useful to fit a NR model. In the case of RR models, they consume bandwidth, which could be used by other services. Finally, we have selected the NR algorithms, because they do not consume bandwidth and require a minimum amount of calculation.

The work of M. Venkataraman and M. Chatterjee [5] is a good starting point as they try to estimate the quality without any reference of the original video. However, it is not very portable, because the application needs to access a database constantly and this may lower the efficiency of the application and requires a large storage media. There are some other works [6]–[8] that also propose ways to translate a set of network measurements to QoE with NR algorithms. Our contribution is to complement them with the study on how MPEG frame types losses affect the QoE.

III. METHOD DESCRIPTION

This section describes the method used to obtain our model.

A. Video Setup

For our experiments, we use the video files proposed at [9], taking a set of 23 videos. These video sequences were selected since they are representative of different levels of spatial and

temporal complexity, as computed by means of the Spatial Information (SI) and Temporal Information (TI) indexes. The sequences have CIF and QCIF resolution at 25 fps. We encoded all sequences using H.264 and MPEG-TS. We have used these video sequences to let our work be compared with previous related works.

B. Experiment Data

The developed model is based on the following experiment data. For each of the 23 original H.264 videos, we generated a number of corrupted video streams by dropping packets according to a given packet loss rate (PLR). To simulate burst errors, we have used the Netem [11] Linux patch to generate six different PLR in the set $\{0.1\%, 0.4\%, 1\%, 3\%, 5\%, 10\%\}$ with a two state Gilbert-Elliot's [10] model with an average burst length of 3 packets, since it is characteristic of IP networks [12], [13].

For each PLR and content, 10 realizations were produced, thus obtaining a total of 1380 distorted files.

C. VQM General Model

To calculate the QoE of the distorted videos, we have used the VQM General Model [14]. This model was evaluated by the Video Quality Experts Group (VQEG), being the only video quality estimator that was best ranked for several video tests.

We have used the VQM General Model with our 1380 distorted videos. The obtained results were expressed in the VQM scale, so a map was needed to express the result in the MOS scale. VQM values were mapped to MOS scale as shown in (1):

$$MOS = 5 - 4 \cdot VQM \quad (1)$$

D. Network Monitoring

To assess each video, we developed an application that captures each video stream and measures a set of network parameters. We split the application into two modules, the first one is responsible for capturing the contents of the network traffic, splitting it in different video flows, and marking each packet with its collection time for further analysis. The second module is responsible for analyzing the contents of each captured flow and translating it into a set of preselected IPFIX records [15], [16]. We chose ten parameters to measure the state of the network (rtpPacketCount, rtpPacketCountLoss, rtpJitterSum, etc.) and thirteen parameters to identify the record (rtpSSRC, rtpFlowId, destinationIPv4Address, etc.).

To create these records we have inspected each RTP message to count the number of received packets, lost packets and discarded packets because of a late arrival to the decoder. We have also measured packet inter-arrival time and its jitter. Each RTP message contains an MPEG stream, where a GOP (Group of Pictures) header and the MPEG picture header can also be analyzed to obtain the type of MPEG-frame, frame size and frame temporal sequence.

These records are dumped to a file to be later analyzed.

IV. QOE PREDICTION MODELS

This section provides the achieved results with the obtained data. All regression data have been calculated primarily through the Smith's Statistical Package [17].

A. Packet Loss Rate Based Model

Figure 1 shows the MOS values of the distorted videos as a function of the PLR. The MOS trend is not clear, but it can be observed that an increase of packet losses gives a lower average MOS.

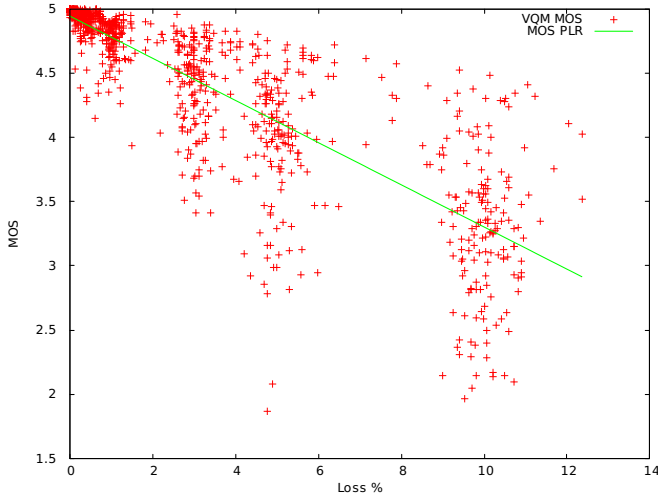


Fig. 1. General Model VQM MOS vs PLR model

Using a simple linear regression, we can approximate the real MOS value. We tried with three different functions: exponential, logarithmic and linear. The function with the best goodness-of-fit was the linear function, as expressed in (2).

$$MOS(plr) = a + b \cdot plr \quad (2)$$

The coefficients resulting of the linear regression are shown in table I.

TABLE I
RESULTS OF PLR LINEAR REGRESSION

Variable	Coefficient	Std. Error	t Value	One-sided p-value
a	4.9442	0.0165	299.4776	0
b	-0.1642	0.0034	49.6947	0
SEE	0.3757			
R^2	0.6974			

This model has a mean squared error equal to 0.1412 with respect to the VQM general model, which gives a mean absolute error of 0.3757 (9.38% in the MOS scale). This is not a significant error if we consider that the algorithm does not have any reference of the original video. However, it can be seen that a linear behavior does not fit well in the MOS scatter plot. Also, the Pearson squared correlation coefficient R^2 has a value of 0,6974.

B. MPEG Frame Types

To obtain a model that better resembles the MOS behavior, we have looked into MPEG encoding, where the video streams have three different types of frames:

- 1) **I-frame**: it is an intra-coded picture, in summary a conventional picture. All the other types of frames need I-frames to work properly. Then, they are the least compressed frames but they have a very low rate of occurrence.
- 2) **P-frame**: it is a predicted picture. It only contains the changes in the video from the previous I-frame or P-frame, making it highly compressed.
- 3) **B-frame**: it is a bidirectionally predicted picture, using the differences between the previous and following frame. They are the highest compressed pictures.

As it can be imagined, a loss of an I-frame does not have the same weight than a loss of a B-frame. Thus, we have investigated how different frame type losses affect to quality.

C. MPEG Frame Loss Model

Having in mind the different MPEG frame types, we have developed an algorithm to identify them and measure the loss rates for each frame type.

We have found that the videos with a lower MOS had a large I-frame loss rate. This is very reasonable, because a loss in an I-frame implies that B-frames and P-frames that reference to the missing piece may not function properly. Also, if a P-frame is lost, the B-frames that depends of it will be affected too. On the other hand, if a B-frame is lost, any other frame will not be affected. Figure 3 shows the MOS behavior with respect to each MPEG frame type loss ratio.

Based on the idea that the loss of each MPEG frame type is different, we have approximated the MOS function with these three parameters as shown in (3), and compared it with the previously obtained PLR model.

$$MOS(I_{loss}, B_{loss}, P_{loss}) = a + b \cdot I_{loss} + c \cdot B_{loss} + d \cdot P_{loss} \quad (3)$$

Table II shows the coefficient resulting of the multiple linear regression.

TABLE II
COEFFICIENTS AND RESULTS OF MPEG FRAME LOSS MODEL

Variable	Coefficient	Std. Error	t Value	One-sided p-value
a	4.9437	0.0165	298.7312	0
b	-17.5488	1.5542	11.2913	0
c	-0.8657	0.3576	2.4212	0.0078
d	1.6701	0.9601	1.6738	0.0472
SEE	0.3764			
R^2	0.6969			

This model has a mean squared error equal to 0.1416 with respect to the VQM general model, which gives a mean absolute error of 0.3764 (9.5% in the MOS scale). These early results were not very hopeful, because linear, logarithmic and exponential regression had an mean squared error a bit higher

than the previous PLR model. Moreover, R^2 is 0.6969, a bit lower than the PLR model. Even so, figure 2 shows that this model has a scattered behavior more similar to the VQM MOS model than the PLR model.

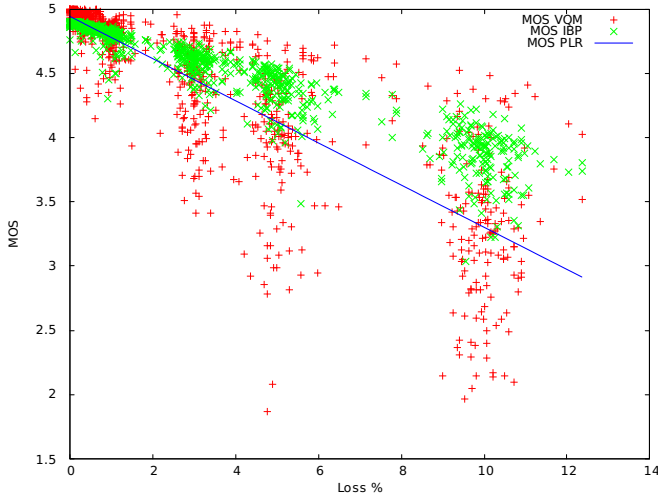


Fig. 2. Initial MPEG frame loss model vs PLR

To improve this model, we used the idea that when the frame header information is lost, then the entire frame is also undecoded. On the other hand, if some frame internal information is lost, the rest of the frame can be decoded. Then, following this idea we modified our data model to measure the different losses in MPEG frame types.

Figures 3 and 4 show the different behavior of MOS depending on the frame loss rates. In both figures, the distribution of points is related to the loss rates of each frame type (taking into account the header losses in the second figure). The focus has to be pointed to the color gradient through them, which represents the MOS values. In the first version of the frame loss ratio measurement low MOS values can be found near to zero. However, in the second algorithm a clearer distribution of colored sectors can be distinguished. This indicates that this second algorithm will be able to better resemble the MOS behavior with the I, B and P-frame loss ratios.

Three functions have been fitted (linear, exponential and logarithmic) to new collected data. We have found that the new linear model (based also on (3)) fits better than both the PLR model and the previous MPEG frame loss model. The multiple linear regression results are provided in table III.

TABLE III
COEFFICIENTS AND RESULTS OF MPEG FRAME LOSS MODEL INCLUDING FRAME HEADER INFORMATION

Variable	Coefficient	Stdr. Error	t Value	One-sided p-value
a	4.9030	0.0142	344.9143	0
b	-1.0823	0.2935	3.6876	0.0001
c	-3.2792	0.4759	6.8911	0
d	-3.2323	0.2784	11.6111	0
SEE	0.3367			
R^2	0.7575			

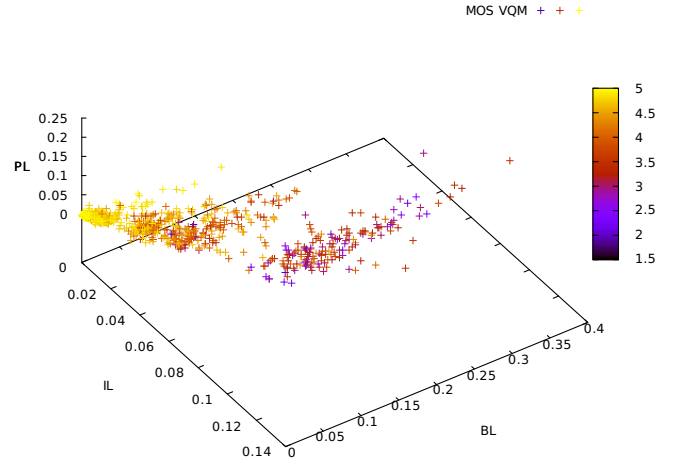


Fig. 3. MOS of different MPEG frame loss rates

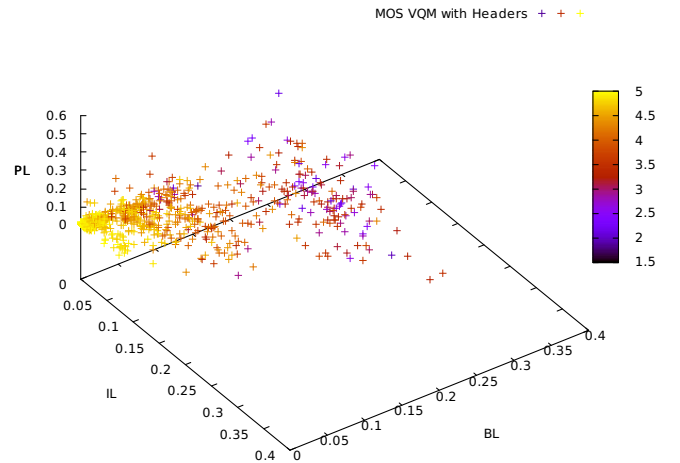


Fig. 4. MOS of different MPEG frame loss rates including frame header information

The new MPEG frame loss model has a mean squared error equal to 0.113 with respect to the VQM general model, which gives a mean absolute error of 0.3367 (8.4% in the MOS scale). Also, R^2 is now 0.7575. Then, these values confirm that this model has a better performance than the previous two estimations. Figure 5 shows how the new MPEG frame model fits better to the VQM model than the PLR model.

It is worth noting that these coefficients are significantly different from those previously calculated. This difference may be due to a number of reasons. The first reason is that most of the decoders are not able to decode a frame if the header is lost. The second is that the header length to frame length ratio is not constant in all three types of frames. Also as we can see the first MPEG frame loss model is not very consistent because P-frame losses increases the video quality obtained. These reasons make the first model unrealistic.

Continuing with the obtained coefficients, they are different

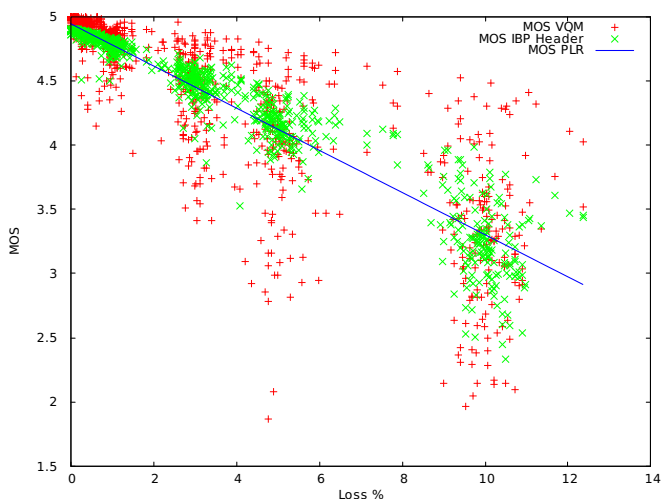


Fig. 5. MPEG frame loss model with headers vs PLR

from what should be expected (greater weight to I-frames, then P-frames and finally lower weight to B-frames), but it should be noted that the loss ratio is obtained with respect to the total number of frames in the video sequence. In our experiment for every 22 frames making up a single GOP structure type there are 1 I-frame, 7 P-frames and 14 B-frames. This means that even with the limited occurrence with i-frames and p-frames, they heavily weigh in the VQM MOS, because other frames depend on them.

V. CONCLUSIONS AND FUTURE WORK

This paper has provided a NR QoE measurement method for IPTV services, trying to minimize the error. For this, we have used the information contained in the MPEG streams.

The cost of this improvement is related to the level of inspection that has to be accomplished. To use the MPEG frame loss model it is necessary to deepen into the MPEG stream transported over RTP. On the other side, when the PLR model is used it is only needed to deepen into the RTP header.

As future work, we will investigate on how the amount of movement also influences on the obtained MOS, and how such movement can be estimated. In high-motion, the importance of any MPEG frame loss will be higher and will affect more to the quality of experience, whereas in a low-motion video, it will affect less to the quality.

Also, it is interesting to consider how well the obtained model fits with an experiment with a panel of users to get the subjective evaluation. Moreover, another database of video sequences can be created to compare the obtained results. Finally, an evaluation of the computational cost of both MPEG frame loss and PLR measurement algorithms can also be done, to decide which one is better to be included in a network probe with a given computational power.

ACKNOWLEDGMENT

This work has been carried out in the framework of the Celtic and EUREKA initiative IPNQSIS (IP Network Monitoring for Quality of Service Intelligent Support) [18] and has been partially funded by CDTI under Spanish PRINCE (PRODUCTO INDUSTRIAL para la gestión de la Calidad de Experiencia) project.

REFERENCES

- [1] J. Gozdecki, A. Jajszczyk, and R. Stankiewicz, "Quality of service terminology in ip networks," *Communications Magazine, IEEE*, vol. 41, no. 3, pp. 153–159, 2003.
- [2] A. Van Moorsel, "Metrics for the internet age: Quality of experience and quality of business," in *Fifth International Workshop on Performance Modeling of Computer and Communication Systems*, vol. 34, no. 13, Nürnberg, Germany, 2001, pp. 26–31.
- [3] M. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *Broadcasting, IEEE Transactions on*, vol. 50, no. 3, pp. 312–322, Sept.
- [4] U. Engelke and H.-J. Zepernick, "Perceptual-based quality metrics for image and video services: A survey," in *Next Generation Internet Networks, 3rd EuroNGI Conference on*, 2007, pp. 190–197.
- [5] M. Venkataraman and M. Chatterjee, "Inferring video qoe in real time," *Network, IEEE*, vol. 25, no. 1, pp. 4–13, January-February 2011.
- [6] M. Fiedler, T. Hossfeld, and P. Tran-Gia, "A generic quantitative relationship between quality of experience and quality of service," *Network, IEEE*, vol. 24, no. 2, pp. 36–41, 2010.
- [7] P. de la Cruz Ramos, J. Navarro Salmerón, R. Pérez Leal, and F. González Vidal, "Estimating perceived video quality from objective parameters in video over ip services," in *ICDT 2012, The Seventh International Conference on Digital Telecommunications*, April 29, 2012, pp. 65–68.
- [8] J. Pokhrel, B. Wehbi, A. Morais, A. Cavalli, and E. Allilaire, "Estimation of qoe of video traffic using a fuzzy expert system," in *Consumer Communications and Networking Conference (CCNC), 2013 IEEE*, 2013, pp. 224–229.
- [9] "Yuv qcif reference videos." [Online]. Available: <http://www2.tkn.tu-berlin.de/research/evalvid/qcif.html>
- [10] G. Hasslinger and O. Hohlfeld, "The gilbert-elliott model for packet loss in real time services on the internet," in *Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB), 2008 14th GI/ITG Conference -*, 31 2008-April 2, pp. 1–15.
- [11] "Netem." [Online]. Available: <http://www.linuxfoundation.org/collaborate/workgroups/networking/netem>
- [12] M. Ellis and C. Perkins, "Packet loss characteristics of iptv-like traffic on residential links," in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, Jan., pp. 1–5.
- [13] M. Borella, D. Swider, S. Uludag, and G. Brewster, "Internet packet loss: measurement and implications for end-to-end qos," in *Architectural and OS Support for Multimedia Applications/Flexible Communication Systems/Wireless Networks and Mobile Computing., 1998 Proceedings of the 1998 ICPP Workshops on*, Aug, pp. 3–12.
- [14] M. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *Broadcasting, IEEE Transactions on*, vol. 50, no. 3, pp. 312–322, Sept.
- [15] J. Quittek, S. Bryant, B. Claise, and P. Aitken, "Information model for ip flow information export," RFC 5102, Jan. 2008.
- [16] H. Scholz and V. GmbH. (2012) Rtp stream information export using ipfix. Internet draft. [Online]. Available: <http://tools.ietf.org/html/draft-scholz-ipfix-rtp-msg-00>
- [17] "Smith's statistical package." [Online]. Available: <http://economics-files.pomona.edu/StatSite/SSP.html>
- [18] "Ipnqsis project." [Online]. Available: http://ipnqsis.org/index.php?title=Main_Page

From: saconet2013-chairs@edas.info

To: diego.hernando@estudiante.uam.es

Date: 26 de abril de 2013 12:30

Subject: [SaCoNeT 2013] Your paper #1569745579 ('Evaluating quality of experience in IPTV services using MPEG frame loss rate') has been accepted

Dear Mr. Diego Hernando-Loeda:

Congratulations - your paper #1569745579 ('Evaluating quality of experience in IPTV services using MPEG frame loss rate') has been accepted by IEEE ComSoc Technical co-sponsorship SaCoNeT'2013 which will be published in the Proceedings of IEEE SaCoNeT 2013 and IEEE Xplore.

Please carefully read the contents below, and follow up them.

Submission of the final version of your paper: The final manuscript is due by April 15 and you must be registered for IEEE SaCoNeT 2013 to upload your paper. The IEEE SaCoNeT 2013 Registration will open on 9 April 2013, detailed instructions will be provided at <http://www.lissi.fr/saconet2013/wiki/registration>. Note that the final manuscripts are to be uploaded through EDAS.

Registration: To be published in the IEEE SaCoNeT 2013 Conference Proceedings and IEEE Xplore, an author (including students) of an accepted paper is required to register for the conference at the FULL rate and the paper must be presented at the conference.

Each paper must be registered with at least one Full Author registration by April 15, 2013 at the latest. For authors with multiple accepted papers, a reduced fee of each additional paper is requested. The full author registration fees include: the welcome cocktail, the coffee breaks, the gala dinner, the digital proceedings (including archiving, publication, and indexing), and the participation to all the Workshops under SaCoNeT 2013.

Accepted and presented papers will be published in the IEEE SaCoNeT 2013 Conference Proceedings and in IEEE Xplore.

Selected papers from those accepted and presented in the conference papers will be considered for publication in Network and Computer Applications Journal (Elsevier Ed., ISI and Scopus Indexed) and Annals of Telecommunication Journal (Springer Ed., ISI and Scopus Indexed).

Paper Revision: Note that each accepted paper is limited to 5 pages. Each extra page (for papers with more than 5 pages) will be charged with 75 over length charge. The final submitted version must be revised according reviewers' comments. However, both the title and author list of an accepted paper can NOT be changed

in the final manuscript. Failure to abide this policy may result in dropping your paper from the program.

No-Show Policy: The organizers of IEEE SaCoNeT 2013 as well as attendees expect accepted papers to be presented at the conference. SaCoNeT reserves the right to exclude a paper from distribution after the conference (e.g., removal from IEEE Xplore) if the paper is not presented at the conference.

We suggest you make your airline and hotel reservations as early as possible. All informations are available for your venue (Hotels, Visa, etc.) at <http://lissi.fr/saconet2013/wiki/venue>.

Looking forward to seeing you in Paris.

Reviewers' Comments: The reviews are displayed below or can be found at <http://edas.info>, using your EDAS user name diego.hernando@estudiante.uam.es.

=====
Review 1
=====

> *** Submission Policy: Does the paper list the same author(s), title and abstract in its PDF file and EDAS registration?

Yes.

> *** Strong aspects: Comments to the author: what are the strong aspects of the paper?

The topic is interesting. No reference metrics for video QoE are attracting a lot of interest at the moment. Comparing to VQM is a reasonable method. The paper is well structured.

> *** Weak aspects: Comments to the author: what are the weak aspects of the paper?

In order to be really interesting, a comparison should be made with existing no reference models. There are several no reference models proposed using more than just packet loss rate. I do not find several of these in the litterature study, which i find surprising.

VQM is mapped linearly to MOS. I believe that there are nonlinear mapping methods proposed and agreed upon, that provide better fit, see e.g. ITU standards.

You say that you created 1380 distorted files, but later you say that you use the general model on your 5520 distorted videos. This is confusing. But the main problem i have with the paper is that it seems like the training of the model and the evaluation of it, is performed on the same dataset. A common, and scientific way to do this, is to use a dataset for training of the model, and to evaluate it on a completely different dataset. Otherwise the results will not reliable.

Furthermore, the evaluation of the proposed model is too simple. It is common to evaluate models against MOS using Pearson or Spearman correlation. I would like to see that here too. So, the contribution is limited. The authors simply state that the model is better than using just packet loss. This is where a comparison with existing models would be appropriate.

> *** Recommended changes: Recommended changes. Please indicate any changes that should be made to the paper if accepted.

I suggest to use different datasets for training of the model and for evaluation of the model.

I suggest to evaluate the model statistically using correlation tests.

I suggest to include references to existing no reference models, and a comparison with one of them.

> *** Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.

Good (4)

> *** Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.

Marginal work and simple contribution. Some flaws. (2)

> *** Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.

Minor variations on a well investigated subject. (2)

> *** Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.

Readable, but revision is needed in some parts. (3)

=====
Review 2
=====

> *** Submission Policy: Does the paper list the same author(s), title and abstract in its PDF file and EDAS registration?

Paper lists the same authors, title and abstract in its PDF file and EDAS registration.

> *** Strong aspects: Comments to the author: what are the strong aspects of the paper?

The paper addresses interesting question of NR video quality estimation based on MPEG frame loss. The paper reads generally nicely and the approach used is clearly presented.

> *** Weak aspects: Comments to the author: what are the weak aspects of

the paper?

The authors have decided to trust VQM model as the ground truth for their work. There is no visible consideration of other options and discussion on if this decision yields to some bias in the presented results.

The authors have decided to model packet loss - QoE connection with simple linear regression (also exponential and logarithmic were tried). It is not discussed that it is possible that packet loss - QoE connection may be too complex to be modeled accurately by such a simple way.

The paper focuses QoE model for IPTV services. Still, the video clips used are CIF and QCIF resolution, which are quite much lower than the normal IPTV resolutions. Authors should explain their decision on this.

> *** Recommended changes: Recommended changes. Please indicate any changes that should be made to the paper if accepted.

The idea of taking the MPEG frame into account in QoE estimation is good. This must come with some kind of an extra requirement for computing needed. The authors could add a short consideration on what is the meaning of this extra processing need keeping in mind the goal of being able to use general purpose computers.

Claim "...every network provider has in its portfolio.." is not true.

First sentence of III needs to be reformulated.

Add legend to Fig 1, if different colors used have specific meaning.

Check Fig1 caption.

III.B. refers to 1380 video files, III.C. to 5520 videos. It is not clear

why the latter number is 4-fold.

End of IV.A: "it does not resemble the real MOS behavior". How do you measure resemblance? Same question applies to the statement above Fig 3.

The three dimensional figures 4 and 5 are hard to read. If possible, try to make them more clear or consider presenting two dimensional sections.

There is a great difference coefficients (d even has effect two opposite directions) presented in tables 2 and 3. Can you discuss on the possible reasons and consequences?

Check reference formats. E.g. [5] misses publication year, [7] misses conference name.

Proofread the whole paper for small errors.

> *** Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.

Excellent (5)

> *** Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the analysis or simulation study, thoroughness of the treatise, accuracy of the models, etc.), its soundness and scientific rigour.

Valid work but limited contribution. (3)

> *** Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.

Some interesting ideas and results on a subject well investigated. (3)

> *** Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.

Readable, but revision is needed in some parts. (3)

=====
Review 3
=====

> *** Submission Policy: Does the paper list the same author(s), title and abstract in its PDF file and EDAS registration?

Yes

> *** Strong aspects: Comments to the author: what are the strong aspects of the paper?

The authors have defined a model to predict the experienced quality that is a function of loss of the different types of MPEG frames

> *** Weak aspects: Comments to the author: what are the weak aspects of the paper?

Benchmarking with other models to predict the experienced quality evaluations results.

> *** Recommended changes: Recommended changes. Please indicate any changes that should be made to the paper if accepted.

Figure 1 has low resolution, please include a new one with higher resolution.

> *** Relevance and timeliness: Rate the importance and timeliness of the topic addressed in the paper within its area of research.

Excellent (5)

> *** Technical content and scientific rigour: Rate the technical content of the paper (e.g.: completeness of the analysis or simulation study, thoroughness of

the treatise, accuracy of the models, etc.), its soundness and scientific rigour. Solid work of notable importance. (4)

> *** Novelty and originality: Rate the novelty and originality of the ideas or results presented in the paper.

Some interesting ideas and results on a subject well investigated. (3)

> *** Quality of presentation: Rate the paper organization, the clearness of text and figures, the completeness and accuracy of references.

Well written. (4)

Best regards,
Abdelhamid Mellouk
General chair

Nadjib Aitsaadi, Xavi Masip-Bruin, Mohammed Younis
TPC co-chairs

B

Interfaz web del sistema de medidas

En este anexo se adjunta el trabajo realizado sobre la maqueta hecha para mejorar la página web disponible para la consulta de la calidad de experiencia mostrada en el capítulo 6, así como el servicio web necesario para la recepción de datos.

Esta página deberá tener por debajo una base de datos que guarde la información de los registros recibidos y de los usuarios del sistema. En esta base de datos deberán existir dos tipos de usuarios, los que puedes realizar inserciones en las tablas y los que solo podrán ver el contenido de las deseadas.

Además se deberá añadir un módulo de comunicación con la aplicación principal, poniendo a su disposición un servicio web a través del cual comunicarse. Este servicio será el encargado de volcar la información obtenida a través del servicio web a la base de datos.

Por otro lado, se ha elegido para esta versión preliminar una interfaz *Html* básica, aunque se podrá ampliar con una serie de complementos como *JQuery*, *Ajax*, etc. En la figura B.1 se puede ver la pantalla de acceso a la aplicación web. Al acceder a través de esta pantalla se mostrarán una serie de opciones en las siguientes, dependiendo de si el usuario que ha hecho *log in* es administrador o un usuario normal.

En la pantalla de inicio, que se puede ver en la figura B.2, podemos comprobar la existencia de las opciones “Registro de capturas”, “Gráficas”, disponibles para todos los usuarios, y “Administración de usuarios”, disponible para los usuarios de tipo Administrador.

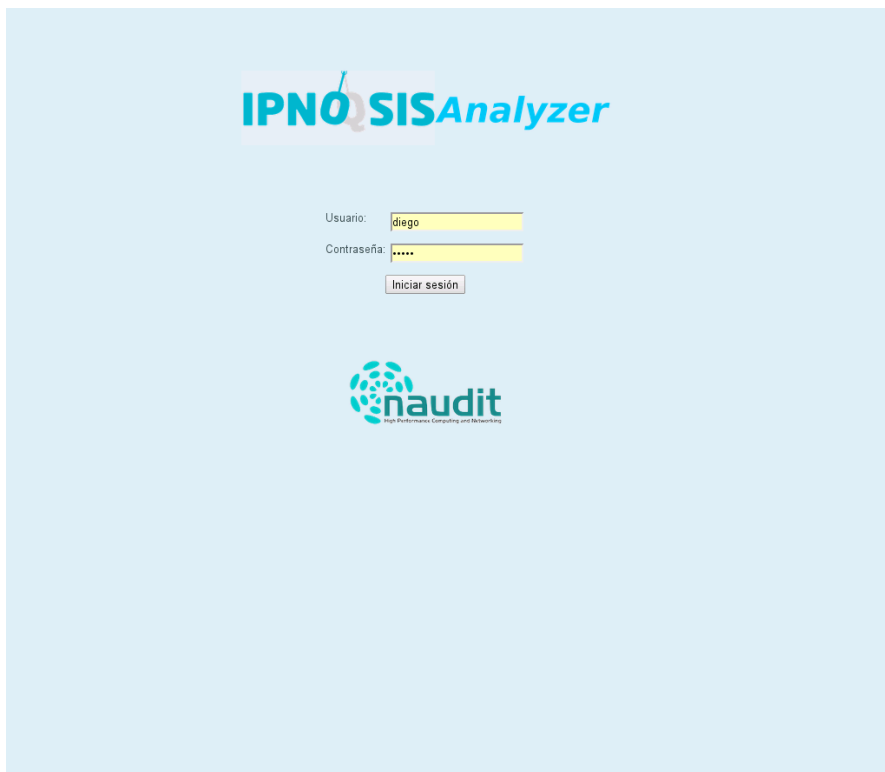


Figura B.1: Pantalla de acceso

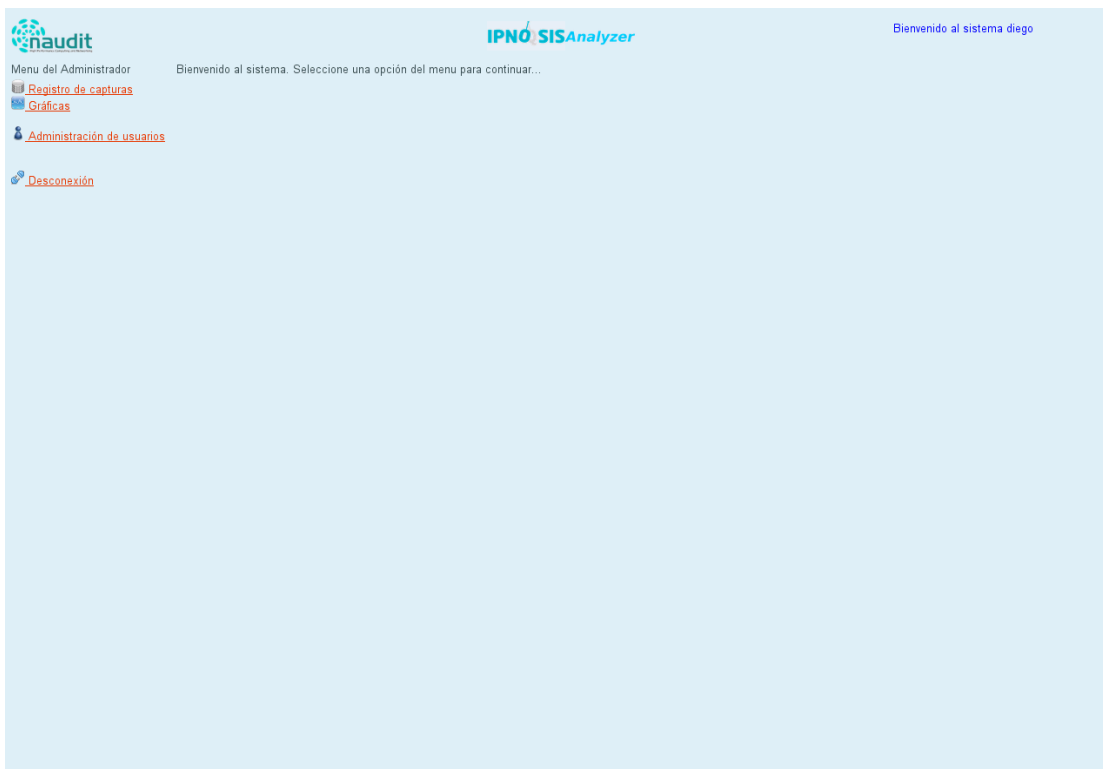


Figura B.2: Página de inicio

En la figura B.3 se puede ver la ventana de “Registro de capturas”, en la que se tendrá acceso de forma visual a los registros almacenados en la base de datos. En esta ventana se podrán aplicar filtros para facilitar la búsqueda de registros y se podrán almacenar para hacer más cómodas las búsquedas. Además, se podrán ordenar los datos por el campo que se especifique.

The screenshot shows the IPNO SISAnalyzer web interface. On the left is a navigation menu with options like 'Registro de capturas', 'Gráficas', 'Administración de usuarios', and 'Desconexión'. The main area displays a table of network capture records. A 'Filtrado de trazas' dialog box is open, allowing users to filter records based on date range, IP addresses, ports, and MOS. The table below the dialog shows records with columns: id, sourceIPv4Address, sourceIPv4AddressStr, protocolIdentifier, sourceTransportPort, sourceIPv4Address, sourceIPv4AddressStr, and destinationTransportPort.

id	sourceIPv4Address	sourceIPv4AddressStr	protocolIdentifier	sourceTransportPort	sourceIPv4Address	sourceIPv4AddressStr	destinationTransportPort
71	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
70						10.0.104.48	16083
69						10.0.104.48	16083
68						10.0.104.48	16083
67						10.0.104.48	16083
66						10.0.104.48	16083
65						10.0.104.48	16083
64						10.0.104.48	16083
63						10.0.104.48	16083
62						10.0.104.48	16083
61						10.0.104.48	16083
60						10.0.104.48	16083
59						10.0.104.48	16083
58	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
57	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
56	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
55	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
54	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
53	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
52	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
51	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
50	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
49	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083
48	3379036170	10.0.104.201	17	32821	812122122	10.0.104.48	16083

Figura B.3: Registros almacenados

En la figura B.4 se puede ver la ventana “Gráficas”, en la que se mostrarán una serie de gráficas seleccionadas por el usuario que tomarán los datos de cualquier campo de los registros de la base de datos. Este campo se podrá seleccionar pinchando sobre la gráfica, donde se desplegará un menú donde se podrá seleccionar el campo deseado (MOS, jitter, delay, etc.). Además, se podrá especificar el intervalo de tiempo de visualización de datos. En la figura, se puede ver un ejemplo de una visualización de la evolución del MOS sobre un canal.

Finalmente, en la figura B.5 se puede ver la ventana de “Administración de usuarios”, donde los administradores podrán modificar los permisos de cada uno de los usuarios.



Figura B.4: Gráficas



Figura B.5: Gestión de usuarios

