Implementación y evaluación de la codificación LDPC para la transmisión de ficheros en entornos unidireccionales

Autor: Ismael de Fez Lava
Director: Dr. Juan Carlos Guerri Cebollada

Resumen — La utilización de mecanismos FEC (*Forward Error Correction*) proporciona una mayor fiabilidad en la transmisión de contenido IP en entornos con pérdidas. De entre los distintos tipos de códigos FEC existentes, esta tesina presenta una completa evaluación sobre la codificación LDPC (*Low Density Parity Check*) basada en una implementación propia de dicha codificación, según las características definidas por la RFC 5170. Dicha evaluación muestra las ventajas que supone el empleo de codificación a nivel de paquete en la transmisión de ficheros en entornos unidireccionales, y realiza una comparación entre dos tipos de estructuras LDPC: Staircase y Triangle.

Abstract — FEC (Forward Error Correction) mechanisms improve IP content transmission reliability in the event of packet loss. Among different types of existing FEC codes, this dissertation presents a thorough evaluation of LDPC (Low Density Parity Check) coding, based on an implementation developed within this research project, according to the specifications defined by RFC 5170. The evaluation shows the advantages that packet level encoding provides to file transmission in unidirectional networks, and performs a comparison between two kinds of LDPC structures: Staircase and Triangle.

Autor: de Fez Lava, Ismael, email: isdefez@iteam.upv.es
Director : Guerri Cebollada, Juan Carlos, email: jcguerri@dcom.upv.es
Fecha de entrega: 24-03-2010

ÍNDICE

I. INTRODUCCIÓN Y OBJETIVOS

I.1. INTRODUCCIÓN

En los últimos años se ha producido un auge en la utilización de redes inalámbricas en la industria de las comunicaciones. No hay más que ver la cantidad de tecnologías que han aparecido, como Wi-Fi, DVB, UMTS, WiMAX, *bluetooth*… La sociedad ha ido adquiriendo cada una de estas tecnologías en su vida diaria. Por poner un ejemplo, la gente utiliza UMTS para hablar por el móvil, se conecta a Internet a través de Wi-Fi y ve la televisión por DVB. Esto implica que los usuarios sean cada vez más exigentes y esperen una mayor calidad de los servicios que reciben. Precisamente, uno de los aspectos clave en el diseño de estas tecnologías es la fiabilidad que proporcionan en la transmisión.

Para mejorar las comunicaciones resulta necesario utilizar mecanismos de corrección de errores, ya que en las redes inalámbricas se producen pérdidas en la transmisión. Los sistemas de corrección de errores aparecieron prácticamente con el nacimiento de las telecomunicaciones. En un medio de transmisión, ya sea por cable o inalámbrico, siempre se producen errores en el canal, por lo que la existencia de herramientas que detecten y, sobre todo, corrijan dichos errores es algo esencial en cualquier sistema de comunicación.

Aparte de errores a nivel de bit, se pueden producir errores a nivel de paquete, es decir que haya pérdidas en el canal que propicien que un paquete enviado no sea recibido. Para proporcionar protección frente a esta clase de errores, más allá de las técnicas de detección de errores como ARQ (*Automatic Repeat Request*), se emplean mecanismos FEC (*Forward Error Correction*), que permiten reconstruir un paquete que no se ha recibido. FEC se utiliza principalmente en entornos unidireccionales, en los cuales no existe un canal de retorno, o en sistemas en tiempo real en los que la retransmisión de paquetes no resulta válida. La utilización de este tipo de técnicas conlleva una disminución del tiempo de recepción de un contenido y provoca una reducción del tráfico en la red, ya que evita la solicitud de paquetes perdidos (lo cual no es siempre posible dependiendo del tipo de canal).

Existen distintas categorías principales de FEC: los códigos convolucionales, los códigos bloque, los códigos fuente (como los Raptors) y los sistemas híbridos. En los códigos bloque existen diferentes codificaciones, entre las más conocidas se encuentran Reed-Solomon y la recuperada codificación LDPC, nacida en los años 60.

Precisamente, esta tesina se centra en el análisis, implementación y evaluación de la codificación *Low Density Parity Check*.

I.2. OBJETIVOS DE LA TESINA

Los principales objetivos de la presente tesina son los siguientes:

- Estudio del protocolo de transmisión de contenido FLUTE y de su arquitectura: análisis de las principales características de FLUTE y de los bloques que forman su pila de protocolos.
- Estudio de los diferentes códigos FEC soportados por FLUTE: análisis de Reed-Solomon, Raptors y LDPC, todos ellos codificadores a nivel de paquete.
- Análisis de la codificación LDPC (*Low Density Parity Check*) y de dos tipos de estructuras existentes: Staircase y Triangle.
- Implementación de un codificador y un decodificador LDPC con estructuras Staircase y Triangle, basados en la RFC 5170 [1].
- Evaluación de las ventajas que implica la utilización de codificación FEC en la transmisión de ficheros.
- Evaluación del códec LDPC implementado y comparación entre las dos estructuras existentes en función de distintos criterios (tasa de codificación, tamaño y pérdidas).
- Evaluación del códec en entornos inalámbricos.

I.3. ESTRUCTURA DE LA MEMORIA

Esta tesina está estructurada en seis capítulos principales más una serie de anexos que incluyen las publicaciones en las que el autor ha participado. Además, también se incluye un glosario de acrónimos utilizados. El primer capítulo introduce la tesina y presenta sus principales objetivos.

El segundo capítulo de esta memoria se centra en la transmisión de ficheros en redes inalámbricas multicast. Para ello, se comenzará realizando un breve análisis sobre las distintas redes de difusión que existen en la actualidad para luego profundizar en FLUTE [2]. Este protocolo permite el envío de ficheros en entornos unidireccionales proporcionando fiabilidad en la transmisión, por lo que resulta especialmente apropiado para las redes comentadas anteriormente.

Tras explicar las principales características y el funcionamiento del protocolo, se describirán los diferentes bloques que forman la arquitectura de FLUTE. Uno de estos bloques es el de corrección de errores (FEC), que será descrito en el capítulo siguiente.

El tercer capítulo describe los códigos FEC y presenta las diferentes codificaciones soportadas por FLUTE: Reed-Solomon, Raptors y LDPC. Es esta última codificación la que centra el trabajo de esta tesina.

La implementación de la codificación LDPC se describe en el capítulo cuarto. Las especificaciones de este tipo de codificación están descritas en la RFC 5170, que será explicada en la introducción del capítulo. Después se describen las librerías desarrolladas tanto en la parte del servidor como en la del cliente para implementar el códec LDPC.

El capítulo quinto se centra en la evaluación del códec LDPC implementado. A lo largo de diferentes estudios se demuestran las ventajas que supone la utilización de mecanismos de codificación en la transmisión de ficheros, y se analiza el comportamiento de LDPC en determinadas situaciones. Se realizan para ello diversos análisis en función del tamaño del fichero, de la tasa de codificación utilizada o de las pérdidas del canal, entre otras. Asimismo, se comparan las dos clases de estructuras LDPC implementadas: Staircase y Triangle.

Para cerrar la memoria, el capítulo 6 recopila las principales conclusiones obtenidas a lo largo de este trabajo y se explican los próximos pasos a dar en el estudio y mejora de la codificación LDPC.

II. TRANSMISIÓN DE FICHEROS EN REDES INALÁMBRICAS MULTICAST

II.1. REDES INALÁMBRICAS MULTICAST

Actualmente existen muchos tipos de redes inalámbricas de difusión estandarizadas. Entre las diversos organismos de estandarización y familias existentes podemos encontrar a DVB, 3GPP y Mobile IPTV, entre otros.

DVB (*Digital Video Broadcasting*) es una organización cuyo objetivo es la creación de estándares técnicos de televisión digital y de servicios de difusión de datos. Dichos estándares se utilizan, sobre todo, en Europa. DVB ha definido distintos sistemas que regulan la distribución de contenido por satélite (DVB-S), cable (DVB-C), televisión terrestre (DVB-T) y por televisión terrestre para dispositivos móviles (DVB-H).

Estos estándares son ampliamente utilizados. Por ejemplo, la televisión digital en España sigue el estándar DVB-T. Actualmente existe una evolución, DVB-T2, que ofrece una mejor eficiencia. Otro estándar a destacar es DVB-H [3], que define la difusión broadcast de contenido multimedia a terminales móviles y funciona sobre IP multicast.

Dentro de la amplia gama de estándares que componen el 3G destacamos los estándares de difusión broadcast MBMS (*Multimedia Broadcast and Multicast Services*) [4] e IMB (*Integrated Mobile Broadcast*) [5]. El primero de ellos permite el despiegue de IP multicast sobre redes 3G, ofreciendo, de esta forma, una mejor escalabilidad en la distribución de contenido. Su funcionamiento se basa en la creación de árboles multicast en la red troncal y en que los usuarios comparten los recursos radio en cada nodo de la red.

IMB es un nuevo estándar 3GPP de vídeo que permite el envío de servicios broadcast de forma espectralmente eficiente. Su objetivo es acelerar la adopción del servicio de datos en los dispositivos móviles, y ofrece grandes ventajas a los operadores, que pueden ofrecer sus servicios en el espectro que IMB tiene asignado. Esta tecnología reutiliza aspectos de otras tecnologías como

UMTS y MBMS. Sin embargo, a diferencia de MBMS, IMB no utiliza IP multicast, sino que emplea broadcast.

Tanto IMB como MBMS utilizan el espectro 3G, el cual ya está regulado y en el que su explotación no requiere de nuevas licencias, al contrario de lo que sucede en DVB.

La última red de difusión señalada es Mobile IPTV [6]. Esta tecnología permite el envío y recepción de contenido multimedia (como televisión digital o servicios de voz) a través de redes de cable e inalámbricas basadas en IP. Proporciona calidad de servicio, seguridad, movilidad y funciones interactivas. Mobile IPTV extiende las funcionalidades del estándar de televisión IPTV para adaptarlas a los dispositivos portátiles.

A la hora de transmitir un contenido, si éste es un fichero no se pueden producir errores en la transmisión. Para que un receptor pueda reconstruir correctamente un fichero es necesario recibir todos y cada uno de los paquetes que lo componen. Es por ello, que las redes de difusión vistas necesitan garantizar que el contenido que difunden llega correctamente a los receptores. El problema es que este tipo de redes, por ser casi todas unidireccionales, carecen de mecanismos que proporcionen seguridad en la transmisión (como puedan ser confirmación de paquetes de TCP o la retransmisión selectiva). Resulta, pues, necesario el uso de un protocolo que garantice que en la transmisión no se produzca ningún error o, en su caso, que, aunque se produzcan errores, el receptor sea capaz de recuperar el paquete original. El protocolo utilizado para los servicios de descarga de contenido a móviles por los diferentes organismos de estandarización es FLUTE.

II.2. FLUTE

II.2.1 Descripción

FLUTE (*File Delivery over Unidirectional Transport*), definido en la RFC 3926 [2], es un protocolo para el envío unidireccional de ficheros sobre Internet, especialmente adecuado para redes multicast.

Entre sus principales características destacamos:

- Fiabilidad en la transmisión: a través del envío de contenido en carruseles dinámicos y mediante técnicas de corrección de errores FEC.
- Escalabilidad masiva: esto significa que pueden existir muchos receptores, que el tamaño de los objetos puede variar en una sesión desde kilobytes a gigabytes, que cada receptor puede iniciar la recepción de un objeto asíncronamente, que la tasa de recepción de cada receptor puede ser distinta y que todo esto puede ser soportado utilizando un único emisor.
- Ofrece gestión y control de la congestión mediante el protocolo ALC e IP multicast.
- Es útil para el envío de metadatos, de hecho DVB-H emplea FLUTE para el envío de la Guía Electrónica de Servicios (ESG).

FLUTE funciona sobre el protocolo ALC (*Asynchronous Layered Coding*) [7], el cual está diseñado para una distribución multicast ampliamente escalable, que le proporciona el transporte básico a FLUTE. La combinación de FLUTE y ALC permite una distribución asíncrona de ficheros desde un emisor a múltiples usuarios, lo que la convierte en adecuada para sistemas de distribución masiva en los cuales no hay un canal de retorno (o en los que éste es muy limitado).

La transmisión del contenido se produce a través de sesiones de envío. Una sesión está unívocamente identificada por la dirección IP multicast donde se envían los contenidos y por un identificador de sesión llamado TSI (*Transport Session Identifier*). Cada sesión contiene uno o más canales de envío asociados. Cada uno de estos canales transmite en un determinado puerto y a una determinada tasa de transmisión.

Cada fichero que se envía a través de los canales de una sesión está identificado por el TOI (*Transport Object Identifier*), un identificador numérico a nivel de fichero.

Ahora bien, ¿cómo sabe un cliente a qué sesión y canal conectarse? Cuando se ha conectado a un canal, ¿cómo averigua qué ficheros se están transmitiendo? ¿De qué forma se envían y se transmiten esos ficheros? Los siguientes apartados dan respuesta a estas preguntas.

II.2.2 Descripción de Sesión

Antes de poder recibir ficheros tiene que haber un establecimiento de la comunicación desde el cliente hasta el emisor. Para que un cliente pueda unirse a un canal es necesario obtener información del emisor para poder conectarse con él. La información requerida puede ser obtenida mediante la Descripción de Sesión. Ésta se puede transmitir de diferentes formas, siendo la más común el uso del protocolo SDP (*Session Description Protocol*) [8], aunque existen distintos métodos *out-of-band*, como cabeceras HTTP/Mime o a través del protocolo SAP.

La Descripción de Sesión debe incluir obligatoriamente los parámetros que identifican a una sesión: la IP del emisor y el TSI. Asimismo, también incluye información sobre el número de canales, tasa de transmisión y puerto de cada canal, el control de la congestión utilizado, y puede incluir otra información adicional.

II.2.3 Tabla de Transferencia de Ficheros (FDT)

Una vez los clientes disponen de la información necesaria para unirse a una sesión y han establecido la conexión, ya pueden empezar a recibir a ficheros. Pero antes, deberán conocer qué ficheros se están transmitiendo por el canal y sus características. Esta información se obtiene a través de la Tabla de Envío de Ficheros (FDT).

La FDT (*File Delivery Table*) proporciona un medio para describir varios atributos asociados con los ficheros que se envían en la sesión. Los atributos más importantes son: el identificador del objeto (TOI), la localización y nombre del fichero (especificado por el URI), la longitud del fichero o la codificación, entre otros.

La FDT está escrita en lenguaje XML. La figura 1 muestra un ejemplo de una FDT, donde se observa cómo la información está organizada a través de elementos y atributos. Por ejemplo el elemento “File” identifica a un fichero concreto, con una serie de características definidas a través de distintos atributos como “TOI” o “Content-Location”.

El envío de la FDT se produce a través de Instancias FDT, que son paquetes de FLUTE con una extensión de cabecera de FDT. El XML propiamente dicho es el que forma la carga del paquete a enviar. Para identificar un paquete como FDT, se reserva el valor de TOI igual a cero. Si se desea conocer mejor la sintaxis de las FDT, así como su envío a través de paquetes, se recomienda consultar la RFC 3926 [2].

```
<?xml version="1.0" ?>
<FDT-Instance xmlns="http://comm.iteam.upv.es/dvb-h/fdt"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://comm.iteam.upv.es/dvb-h/fdt FDT.xsd"
     Expires="3499186486" Complete="true" FEC-OTI-FEC-Encoding-ID="0"
     FEC-OTI-Maximum-Source-Block-Length="64" FEC-OTI-Encoding-Symbol-
     Length="1428">
      <File TOI="1" Content-Location="file:///ej3.ppd"
        Content-Type="ppd" Content-Length="19186"
        Content-MD5="5jglLZflhb5iRDMXPeTYaQ==">
          <Group>"presentation"</Group>
          <Group>"COMM"</Group>
       </File>
</FDT-Instance>
```

Fig.1. Ejemplo de FDT.

II.2.4 Arquitectura

La figura 2 muestra la pila de protocolos utilizada por FLUTE. En la parte superior se encuentra el protocolo ALC, que le proporciona el transporte básico a FLUTE. A su vez, éste emplea el bloque LCT para funciones de gestión de sesión, un bloque de control de la congestión (CC), así como el bloque FEC encargado del control de errores. En las capas inferiores, FLUTE funciona sobre UDP en el nivel de transporte y utiliza IP en el nivel de red.

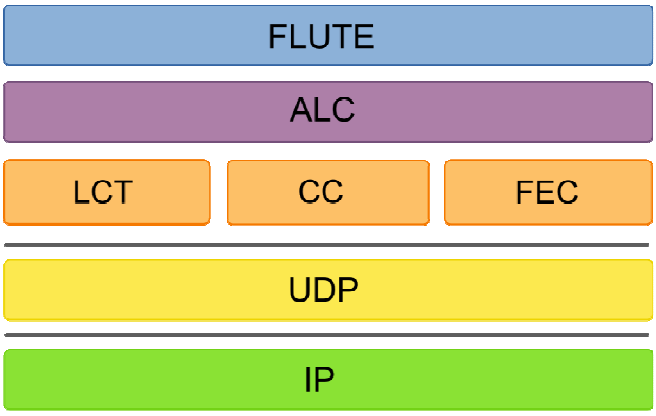


Fig.2. Pila de protocolos de FLUTE.

El bloque de LCT (*Layered Coding Transport*), definido en la RFC 3451 [9], proporciona soporte a nivel de transporte para protocolos ampliamente escalables utilizando IP multicast. Concretamente, da soporte a dos tipos de aplicaciones: transferencia fiable de contenido y aplicaciones de *streaming*. Es un protocolo unidireccional. La cabecera de LCT es la utilizada por FLUTE para el envío de los paquetes. Los campos más importantes de dicha cabecera son: *TSI* y *TOI* (para identificar a la sesión y al objeto respectivamente) y el campo *CodePoint*, en el que se indica la codificación utilizada. Además, existen distintas extensiones de cabecera, entre las que destacan dos principales: *EXT_FDT* (extensión utilizada para la transmisión de la FDT) y *EXT_FTI* (empleada para pasar los parámetros de la codificación que se utiliza).

El protocolo implementa también un bloque de control de la congestión (CC). Mediante el control de congestión, el emisor es capaz de enviar paquetes a diferentes tasas. Los receptores regulan su tasa de recepción en una sesión ajustando qué conjunto de canales están unidos a cada punto en el tiempo dependiendo del ancho de banda disponible entre el receptor y el emisor, pero independientemente de otros receptores.

El último bloque que falta por ver es el de FEC (*Forward Error Correction*), definido en la RFC 5052 [10]. Este bloque se encarga de especificar los parámetros relativos tanto a las características de la codificación propiamente dichas, como a su transporte. FEC define dos clases de códigos: los códigos completamente especificados y los incompletamente especificados. Todas las codificaciones están identificadas por un valor de *FEC Encoding ID* (por ejemplo Raptor tiene un valor de 1, mientras que LDPC Staircase tiene un valor de 3). Aquellos tipos de códigos incompletamente especificados utilizan, además, un identificador llamado *FEC Instance ID*.

Por otro lado, la RFC 5052 también define los contenedores que transportan la información referida a la codificación:

- FEC Payload ID: este campo identifica un paquete dentro de un objeto, a través del bloque al que pertenece (*Source Block Number*, SBN) y del símbolo que representa dentro de ese bloque (*Encoding Symbol ID*, ESI). Por lo tanto, este contenedor se transmitirá en todos los paquetes de FLUTE, ya que identifica al paquete.
- FEC Object Transmission Information: este campo contiene los parámetros pertenecientes a una determinada codificación, como por ejemplo el tamaño de los bloques y símbolos o la tasa de codificación. Se envía a través de la extensión de cabecera de LCT llamada *EXT_FTI*. La RFC de FLUTE [2] señala que este campo debe enviarse en, al menos, uno de los paquetes que forman parte de un determinado objeto. Si bien la FDT puede contener parámetros referidos a la codificación (aunque es un campo opcional), es a través de la *EXT_FTI* mediante la cual se obtiene la información de codificación necesaria. El formato de esta cabecera tiene una parte común para todas las codificaciones, y otra específica según el tipo de codificación.

Para resumir los bloques vistos, la siguiente figura muestra la estructura de un paquete típico de FLUTE. Como hemos comentado, FLUTE funciona sobre IP y UDP. Por encima de las cabeceras de estos protocolos se envía la cabecera de LCT. Las extensiones de cabecera de LCT no se transmiten en todos los paquetes. Recordemos que la extensión de cabecera de la FDT sólo se incluye en los paquetes FLUTE que envían la FDT, y que la extensión de codificación no tiene que mandarse obligatoriamente en todos los paquetes. Existen otros tipos de extensiones de cabecera definidas por FLUTE. Por encima encontramos la cabecera de *FEC Payload ID*, que identifica al símbolo de codificación que se transmite en la carga.

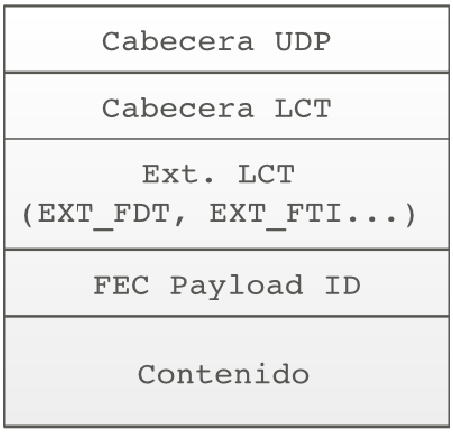


Fig.3. Estructura de un paquete FLUTE.

II.2.5 Transmisión de ficheros

Cada fichero que se envía representa un objeto de transporte, es decir, una combinación de símbolos binarios que forman los datos. Tal como refleja la figura 4, cada objeto de transporte se fragmenta en bloques, según un algoritmo definido por FLUTE. A su vez, cada bloque está compuesto por una serie de símbolos de codificación. Existen dos clases de símbolos de codificación: los símbolos fuente y los símbolos de paridad. Los símbolos fuente son los datos del fichero, mientras que los símbolos de paridad se forman a partir de una combinación entre símbolos fuente para proporcionar fiabilidad en la transmisión. Así, cada bloque contiene n símbolos de codificación, de los cuales k son símbolos fuente. Si se emplea codificación el número de símbolos de paridad por paquete será, por lo tanto, igual a n-k. Cada uno de los símbolos de codificación representa la carga de un paquete FLUTE, al cual se le añade la cabecera correspondiente. También es posible incluir varios símbolos de codificación en un mismo paquete.

En cuanto a la transmisión, hay definidas dos clases de sesiones de envío de ficheros FLUTE:

- Sesión de transmisión de ficheros: el contenido se envía durante un periodo de tiempo determinado y luego se finaliza esa sesión.
- Carrusel de ficheros: en este tipo de sesión los ficheros se envían continuamente a modo de carrusel. Cuando se han transmitido todos los ficheros que forman parte de un canal, se

vuelve a empezar a transmitir desde el principio. De esta forma, los clientes pueden obtener paquetes que todavía no han recibido debido a pérdidas en el canal. Existen dos tipos de carruseles: estáticos y dinámicos. En estos últimos, los ficheros que se envían entre un ciclo y el siguiente pueden cambiar: se pueden añadir, eliminar y actualizar ficheros de la sesión.

Normalmente se suelen utilizar los carruseles como método de transmisión. De hecho, el uso de carruseles, junto a la utilización de FEC es lo que proporciona fiabilidad en la transmisión, que es la característica principal del protocolo FLUTE.

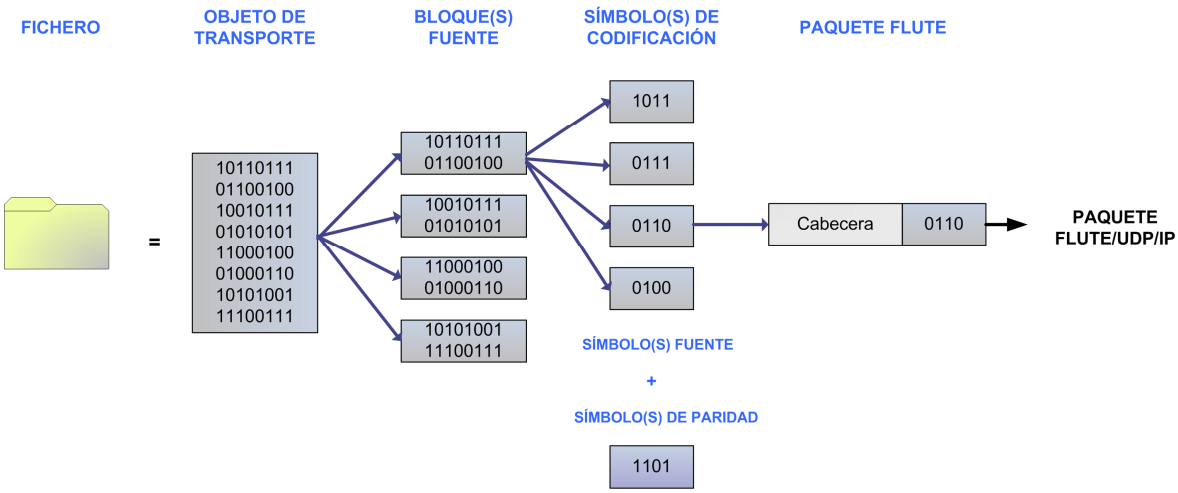


Fig.4. Formación de paquete FLUTE.

III. CÓDIGOS FEC

III.1. INTRODUCCIÓN: TIPOS DE CÓDIGOS

Un tipo de código FEC está definido por los valores de *FEC Encoding ID* y *FEC Instance ID*, tal como vimos en el capítulo anterior. El valor de estos identificadores viene regulado por la IANA (*Internet Assigned Numbers Authority)*. Los *FEC Encoding ID* especificados por IANA son:

0. Compact No-Code: corresponde a no utilizar ningún mecanismo de codificación. Está definido en la RFC 5445 [11].
1. Raptor.
2. Reed-Solomon sobre GF (2^m).
3. LDPC Staircase.
4. LDPC Triangle.
5. Reed-Solomon sobre GF (2^8).

Los valores 6-127, correspondientes a esquemas FEC completamente especificados, están todavía sin asignar. El rango 128-255 está reservado para los códigos FEC incompletamente especificados.

En los siguientes apartados pasamos a detallar las características principales de cada uno de los códigos enumerados, haciendo especial hincapié en la codificación LDPC, objetivo de esta tesina.

III.2. REED-SOLOMON

La codificación Reed-Solomon fue inventada por Irving S. Reed y Gustave Solomon en el año 1960 [12]. Esta codificación se utiliza en múltiples aplicaciones, como en el almacenamiento de datos (por ejemplo en el CD o DVD), en comunicaciones inalámbricas (telefonía móvil) o por satélite, en comunicaciones por cable (ADSL) y en televisión digital (DVB emplea Reed-Solomon para corregir errores en la capa física).

Reed-Solomon es un código bloque corrector de errores basado en polinomios, que crea símbolos a través de secuencias de *m* bits. Cada palabra código está formada por *n* símbolos, de los cuales *k* son símbolos fuente y *r* son símbolos de paridad. La relación entre la longitud de la palabra código y el número de símbolos viene definida por: $n=2^m - 1$. Esta codificación es capaz de corregir errores hasta en *r/2* símbolos.

Precisamente, una de las limitaciones de Reed-Solomon viene definida por su estructura, ya que estos códigos proporcionan menor flexibilidad al crear bloques de tamaño fijo. Sin embargo, su eficiencia al codificar ficheros pequeños provoca que sea una codificación muy utilizada puesto que Reed Solomon, al ser un tipo de código MDS (*Maximum Distance Separable*), permite a los receptores reconstruir los *k* símbolos fuente a partir de cualesquiera *k* símbolos recibidos. Por esta razón, Reed-Solomon se engloba dentro de la categoría de códigos FEC de bloque pequeño. El coste en la codificación con bloques grandes resulta prohibitivo, por lo que esta codificación no se emplea a la hora de codificar bloques de gran tamaño.

La RFC 5510 [13], de abril de 2009, define los esquemas FEC para los códigos Reed Solomon sobre GF (2^8) y sobre GF (2^m). La RFC especifica que el valor de *m* varía entre 2 y 16. En ambos casos, la creación de los *n* símbolos a partir de los *k* símbolos que componen un bloque se produce a través de una matriz de generación. Dicha matriz hace uso de en un polinomio que depende de la longitud de los elementos del campo finito *m*.

La principal diferencia entre el esquema con *FEC Encoding ID* igual a 2, RS sobre GF (2^m), y el de *FEC Encoding ID* igual a 5, RS sobre GF (2^8), se encuentra en la estructura del campo *FEC Payload*. Además, Reed-Solomon sobre GF (2^8) especifica el envío de un solo símbolo por paquete. El resto de características (como el proceso de codificación o decodificación) es el mismo en las dos clases de códigos.

III.3. RAPTOR

Los códigos Raptors [14] fueron creados en el año 2001 por la compañía “Digital Fountain Inc.”. Pertenecen a la categoría de los códigos *fountain*, en los que se pueden generar tantos símbolos como se necesiten automáticamente a partir de los símbolos fuente de un bloque, es decir, que no es

necesaria una tasa fija de codificación. A pesar de ser una implementación propietaria, esta codificación ha sido adaptada por multitud de tecnologías. Entre ellas se encuentra el estándar de transmisión a terminales móviles DVB-H.

Al contrario que los Reed-Solomon, los códigos Raptors no tienen limitaciones en cuanto a la cantidad de datos que se protegen, ya que el número de símbolos fuente puede ser muy grande. Además, su algoritmo de decodificación hace que los requerimientos de procesamiento sean mucho menores que en Reed-Solomon.

Su principal característica es que esta codificación permite generar una cantidad ilimitada de información de paridad, a partir de símbolos fuente. Así, en recepción basta con recibir una cantidad de paquetes ligeramente superior al tamaño del fichero original para poder reconstruirlo, independientemente de los paquetes recibidos, por lo que estos códigos resultan muy eficientes. Además, los algoritmos de codificación y decodificación son excepcionalmente rápidos, lo que posibilita su implementación en *software*.

Los códigos Raptors pueden utilizarse tanto en la capa de aplicación como en la de transporte, y están especialmente optimizados para la protección de aplicaciones de *streaming* y de distribución de datos.

El proceso de codificación se divide en dos pasos: en primer lugar se realiza una precodificación (utilizando un algoritmo o una concatenación de algoritmos de codificación), que genera l paquetes de salida a partir de k paquetes de entrada ($l > k$). El segundo paso consiste en la creación de los n símbolos fuente a partir de los l símbolos precodificados ($n > l$), utilizando los códigos LT (*Luby Transform*) [15], un tipo de códigos *fountain*. El funcionamiento de dichos códigos consiste en que, a la hora de crear un nuevo elemento, se muestrea una distribución de pesos devolviendo un entero d que se encuentra entre los valores 1 y k. Se escogen d símbolos de entrada de forma pseudo aleatoria, de manera que el elemento redundante generado se crea a partir de la suma XOR de esos d símbolos. Cada uno de los símbolos se genera independientemente del resto, pudiendo de esta manera formarse un número ilimitado de símbolos.

No hay una implementación única de la codificación Raptor. Existen diferentes codificadores que ofrecen unas prestaciones u otras en función de lo optimizado que sea el diseño de la precodificación y la distribución de pesos del codificador LT.

En la literatura se pueden encontrar multitud de estudios, como [16], que reflejan las ventajas que supone la utilización de este tipo de codificación frente a otras. Tanto para valores de k pequeños como grandes existen codificadores adecuados que pueden trabajar con baja probabilidad de fallo y con ratios de ineficiencia muy bajos.

La forma en la que se generan los símbolos de codificación y el formato de las cabeceras referentes a Raptor están explicadas en la RFC 5053, de septiembre de 2007 [17]. En dicha RFC también se muestran las tablas utilizadas para la generación de los números aleatorios.

III.4. LDPC

La codificación LDPC (*Low Density Parity Check*) fue inventada por Gallager en 1960 [18]. Pero no fue hasta 30 años después, gracias a MacKay y Neal [19], que empezó a utilizarse. La especificación original ha sufrido una serie de mejoras que facilitan su utilización en distintos entornos. Por ejemplo, son la base de las codificaciones Tornado, LT y Raptor, que acabamos de ver, todas ellas implementaciones propietarias.

LDPC pertenece a la categoría de códigos FEC de bloque grande, en los que resulta necesario recibir más de los *k* paquetes que componen un fichero para poder reconstruirlo. Las codificaciones que se engloban dentro de esta categoría resultan convenientes a la hora de codificar ficheros grandes, ya que el coste computacional no crece de forma excesiva.

Los LDPC son códigos bloques lineales sistemáticos basados en una matriz de comprobación de paridad utilizada en el proceso de codificación y decodificación. Dicha matriz define las relaciones entre los distintos símbolos de codificación (símbolos fuente y símbolos de paridad). La matriz está formada por elementos con valores “0” y “1”, y es dispersa, ya que la mayoría de elementos son nulos. A través de dicha matriz el codificador crea los símbolos de paridad a partir de los símbolos fuente (y otros símbolos de paridad ya creados). Asimismo, en recepción, la matriz se emplea para reconstruir símbolos que no se han recibido, mediante los símbolos de codificación ya disponibles. La siguiente figura muestra un ejemplo de matriz de paridad, y establece las relaciones entre los símbolos de paridad y los símbolos fuente.

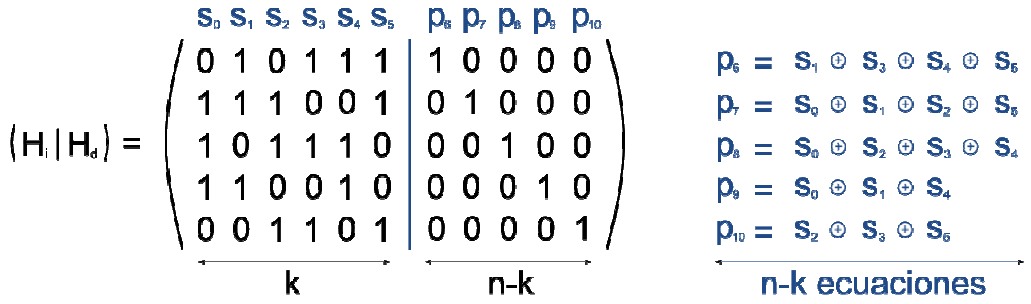


Fig.5. Matriz de comprobación de paridad LDPC.

En la figura 5 se observa una matriz con valores *k*=6 y *n*=11, generándose por lo tanto 5 símbolos de paridad por bloque. El tamaño de la matriz es de *n* x (*n*-*k*), por lo que existen *n* - *k* filas, cada una de las cuales representa una ecuación. Las columnas se corresponden con cada uno de los símbolos del bloque. Cada entrada de la matriz con el valor “1” indica que el símbolo *i*-ésimo participa en la ecuación *j*-ésima. Así, por ejemplo, el primer símbolo de paridad (identificado como p_6), está formado por la suma XOR de los símbolos s_1, s_3, s_4 y s_5. Un símbolo de paridad puede formar parte de la creación de otros símbolos de paridad. Cada uno de los símbolos fuente, normalmente participa en un número fijo de ecuaciones, que corresponde al número de 1s que

contiene esa columna. Ese parámetro se denomina como N1. Al número de elementos no nulos de una fila o columna se le llama grado.

La matriz de paridad está dividida, a su vez, en dos submatrices: la izquierda y la derecha. La primera de ellas está referida a los símbolos fuente, mientras que la submatriz derecha hace referencia a los símbolos de paridad. Al comentar las estructuras LDPC Staircase y LDPC Triangle veremos que ambas variantes tienen la misma matriz H_i, siendo diferente la H_d.

Por lo que respecta al receptor, cuando ha recibido todos los símbolos que componen una ecuación excepto uno, es capaz de recuperar éste haciendo la suma XOR del resto de símbolos de esa ecuación.

Evidentemente tanto el transmisor como el receptor deben utilizar la misma matriz de paridad para que el proceso de decodificación resulte válido. Para la creación de la matriz se emplea un algoritmo definido (en función del tipo de codificación LDPC) que conocen tanto uno como otro. Dicho algoritmo genera la matriz a partir de una serie de parámetros de entrada, estos son: número de símbolos fuente (*k*), números de símbolos de codificación (*n*), número de ecuaciones en las que participa cada símbolo fuente (*N1*) y semilla utilizada para la generación de números pseudoaleatorios (*seed*). Todos estos parámetros los envía el transmisor a través de la cabecera correspondiente. La figura 6 muestra el aspecto que presenta la extensión de cabecera donde se envían dichos parámetros de codificación:

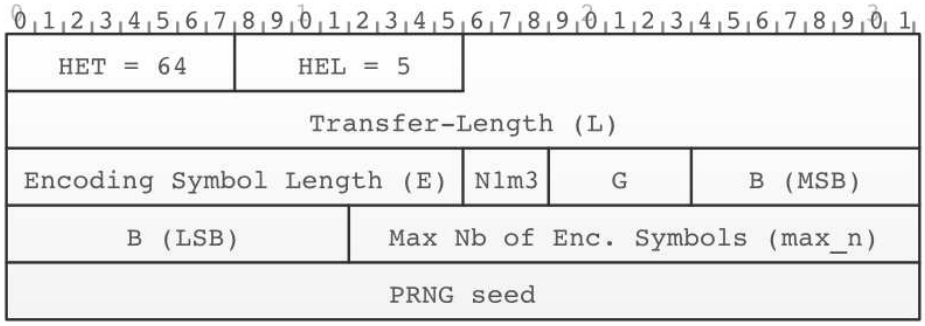


Fig.6. EXT_FTI para *FEC Encoding ID* 3 y 4.

En función de la estructura de la matriz de comprobación de paridad se distinguen dos clases de códigos LDPC: regulares e irregulares. En los regulares, todas las filas de la matriz tienen el mismo grado, y todas las columnas de la matriz tienen el mismo grado. Un código LDPC es irregular si no cumple alguna de estas características. Los códigos de Gallager y Mackay son ejemplos de códigos LDPC regulares, mientras que LDPC Staircase y Triangle son irregulares.

Al respecto, en esta tesina nos centramos en el estudio, implementación y evaluación de estas dos variantes de LDPC: Staircase y Triangle. El nombre de cada una de estas estructuras está definido por la composición de la submatriz derecha H_d, ya que una tiene forma de escalera y otra de triángulo. La figura 7 refleja el aspecto que tiene cada una de estas estructuras:

$$(H_l \mid SC_5) = \left(\begin{array}{cccccc|ccccc} 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \end{array}\right) \qquad (H_l \mid Tr_5) = \left(\begin{array}{cccccc|ccccc} 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{array}\right)$$

Fig.7. Ejemplo de matrices LDPC Staircase y LDPC Triangle (k=6, n=11).

La estructura de la matriz derecha de estas dos variantes de LDPC permite una rápida codificación. A la hora de generar los símbolos de paridad, el símbolo i-ésimo depende únicamente de los *i-1* primeros símbolos. Así, la primera fila de la matriz (que representa la primera ecuación), nos permite hallar el primer símbolo de paridad, ya que se conocen todos los símbolos fuente. Justamente este primer símbolo de paridad forma parte de la segunda ecuación, que al conocer todos los símbolos que forman la ecuación puede generar el segundo símbolo de paridad. Y así sucesivamente hasta la generación del último símbolo de paridad.

Precisamente, esta era una de las principales desventajas de la codificación LDPC original, la forma de la matriz de paridad. En la versión original la matriz estaba formada por una combinación aleatoria de símbolos fuente y símbolos de codificación, por lo que cada símbolo de codificación podía pertenecer a un número aleatorio de ecuaciones. De esta forma, tanto el proceso de codificación como el de decodificación resultaba muy costoso.

Como hemos comentado, la matriz de comprobación de paridad es dispersa, aunque en la figura 7 no se aprecie por ser los valores de *k* y *n* muy bajos. Esta otra figura muestra el aspecto que presenta cada una de las matrices cuando el número de elementos es más elevado.

Fig.8. Ejemplo de matrices LDPC Staircase y LDPC Triangle (k=400, n=600).

Las especificaciones de ambos tipos de estructuras están descritas en la RFC 5170, que se explica a continuación.

IV. IMPLEMENTACIÓN LDPC

IV.1. ESPECIFICACIONES RFC 5170

La RFC 5170 [1], *LDPC Staircase and Triangle FEC*, de Junio de 2008, presenta los códigos completamente especificados (*fully-specified-codes*) con *FEC Encoding ID* 3 y 4, basados en la codificación LDPC Staircase y Triangle, respectivamente. Ambos esquemas pertenecen a la clase de códigos de bloque largos, según la definición de la RFC de FEC [10].

La RFC 5170 define la generación de la matriz de paridad en ambos tipos de estructuras. Para ello, proporciona un algoritmo que, a partir de ciertos parámetros de entrada (como los valores *k* y *n*, o el N1), crea la matriz de paridad. Para ambas estructuras el algoritmo es idéntico a la hora de generar la submatriz izquierda, mientras que difiere en la creación de la submatriz derecha. Para la creación de la matriz la RFC propone el uso del algoritmo generador de números pseudoaleatorios de Park-Miller [20].

Asimismo, la RFC también establece una serie de criterios a la hora de elegir los parámetros de codificación. Por ejemplo, fija un criterio para determinar la longitud máxima del bloque de codificación o la longitud del símbolo de codificación en función de la tasa de codificación utilizada.

Por otro lado, como el transmisor y el receptor trabajan sobre la misma matriz de paridad, hay que transmitir los parámetros necesarios de la codificación a través de la extensión de cabecera *EXT_FTI* (figura 6). En la RFC se definen los campos de esta extensión para ambos tipos de estructuras.

En los siguientes apartados explicaremos el codificador y decodificador LDPC Staircase y Triangle desarrollados en el ámbito de esta tesina. Nuestra implementación respeta los requisitos establecidos por la RFC 5170.

IV.2. DESCRIPCIÓN DE LIBRERÍAS EN TRANSMISIÓN: CODIFICADOR LDPC

La figura 9 muestra la arquitectura del servidor de ficheros implementado, basado en el protocolo FLUTE. Señalar que la implementación del servidor y del receptor de ficheros propiamente dicha a través de FLUTE es resultado de un trabajo anterior del tesitando al realizado en la tesina. En esta tesina se ha implementado específicamente la librería LDPC y los bloques necesarios para realizar los distintos estudios (planificador de paquetes y simulador de pérdidas).

Tanto el transmisor de ficheros como la librería de codificación LDPC están programados en Java. Se ha elegido este lenguaje de programación por su versatilidad y por su compatibilidad con la mayoría de plataformas, principalmente en dispositivos móviles.

El elemento principal corresponde con la librería “jfluteserver”, que implementa el servidor de FLUTE propiamente dicho. Dicha librería gestiona las sesiones, canales y objetos existentes a

través de las clases “FluteSession” y “FluteChannel”. “FluteManager” se encarga de controlar a estas dos clases.

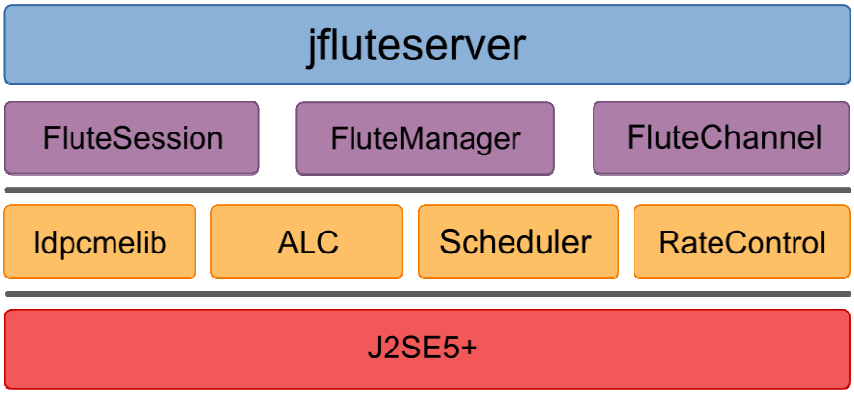


Fig.9. Estructura del servidor de ficheros.

Los paquetes se generan a través del protocolo ALC, que hace uso de la librería “ldpcmelib”, la cual implementa el codificador LDPC y que ha sido desarrollado en el marco de esta tesina. La clase “Scheduler” implementa el mecanismo de planificación de paquetes, mientras que “RateControl” se encarga de enviar el contenido a la tasa indicada por el canal. Todo ello funciona sobre la plataforma Java J2SE5+.

En los siguientes apartados se detalla el proceso llevado a cabo para la implementación del codificador LDPC.

IV.2.1 Creación de matriz de paridad

La librería “ldpcmelib”, utilizada por el servidor y por el receptor, implementa el códec LDPC (tanto la estructura Staircase como la Triangle). Está programada en J2ME, para que pueda ser utilizada en dispositivos móviles. Esta librería se encarga de la creación de la matriz de paridad, que define la relación entre los símbolos fuente y los de paridad.

Para la creación de la matriz se sigue el algoritmo explicado en la RFC 5170. Las estructuras Staircase y Triangle sólo difieren en la parte derecha de la matriz, así que emplean el mismo algoritmo para la generación de la submatriz izquierda.

Como hemos mencionado, la matriz de paridad es dispersa, por lo que para reducir el consumo de memoria se ha implementado una cuádruple lista enlazada por cada fila y columna. Concretamente, la clase principal de la librería es “LDPCMatrix”. Esta clase contiene una instancia de la clase “SparseMatrix”, que es quien implementa la matriz de paridad. “SparseMatrix” contiene dos vectores de tipo “LdpcEntry” que apuntan a las filas y a las columnas de la matriz. La clase “LdpcEntry” representa una entrada en la matriz de paridad, es decir un valor de “1” en la matriz. Cada una de las entradas apunta a las otras entradas vecinas (arriba, abajo, izquierda y derecha). Que sea una cuádruple lista enlazada facilita el proceso de decodificación y reduce el consumo de memoria.

Por otro lado, para la generación de la matriz se utiliza un generador de números pseudoaleatorios, llamado “Park-Miler minimal Standard” (PRNG). Este generador ha sido implementado en la clase correspondiente, y está referenciado por la clase “LDPCMatrix”.

Los parámetros necesarios para la generación de la matriz son (*k*, *n*), que se corresponden con el número de símbolos fuente y el número de símbolos totales de un bloque de codificación. Además, se le pasa el tipo de estructura de la matriz (Staircase o Triangle). Existen otros parámetros opcionales que se pueden introducir que son el N1m3 (número de “1”s por columna en la submatriz izquierda menos 3) y la semilla utilizada para la generación de los números aleatorios. Si no se pasan estos parámetros se utilizan los valores por defecto, 0 y 1 respectivamente.

IV.2.2 Generación de símbolos de paridad

La codificación es por fichero, es decir que cada fichero puede emplear distinta codificación. Es por ello que la generación de los símbolos de paridad se produce dentro de la clase “FluteObject”, ya que son los propios objetos los que se encargan de codificar. Dicha clase representa un objeto FLUTE, que contiene toda la información referida a un fichero, como puede ser el propio contenido, los metadatos o los bloques que lo forman.

Para la división del fichero en bloques y símbolos se utiliza el algoritmo definido en la RFC de FLUTE [2], el cual, recibiendo como argumentos de entrada la longitud del fichero, la longitud del símbolo de codificación y el número máximo de símbolos de codificación por bloque, devuelve el número de bloques en los que se divide el fichero y el número de símbolos fuente que contiene cada bloque. Este algoritmo realiza una división lo más ecuánime posible, para así hacer que todos los bloques tengan aproximadamente el mismo número de símbolos (la mayor diferencia de símbolos entre un bloque y otro es, a lo sumo, de un símbolo).

Una vez obtenido el número de símbolos fuente que forman un bloque (lo denotamos como *k*), se puede calcular el número de símbolos totales (símbolos fuente más símbolos de paridad) que contiene un bloque, en función de la tasa de codificación a emplear. El número de símbolos que forman un bloque lo denotamos como *n*, por lo que el número de símbolos de paridad es *n* - *k*.

Cuando se han hallado estos valores se está en disposición de formar los paquetes. Los primeros *k* paquetes de un bloque contienen datos fuente. Cada uno de estos paquetes está identificado en su cabecera por el número de bloque al que pertenece (*Source Block Number*) y por el número de símbolo que representa dentro de ese bloque (*Encoding Symbol ID*). Los símbolos de paridad se crearán a partir de estos paquetes, en función de la matriz de paridad asociada.

Tras crear la matriz de paridad de ese bloque (pasándole los parámetros *k* y *n*) a través de la librería “ldpcmelib”, se recorren las filas para ir generando los símbolos de paridad. La estructura de la submatriz derecha permite que de cada fila se pueda calcular un nuevo símbolo de paridad, a partir de los símbolos ya disponibles. Así, tras recorrer la fila 1, se podrá crear el símbolo *k+1*, que será la suma XOR de los símbolos que forman parte de la primera fila. De esta forma, la fila *x*

permitirá la construcción del símbolo *k+x*. Cada uno de los símbolos de paridad generados tiene su respectiva cabecera LCT así como su *FEC Payload ID* (con el número de bloque y de símbolo que los identifica).

Asimismo, hay que generar la extensión de cabecera *EXT_FTI* (figura 6), que especifica los parámetros referentes a la codificación. Por lo que respecta a LDPC, los parámetros que forman parte de su cabecera son: longitud de transferencia, longitud de símbolo de codificación, número de entradas por columna de la matriz (N1m3), número de símbolos de codificación por grupo, longitud del bloque fuente máxima, número máximo de símbolos de codificación y semilla para la generación de la matriz de paridad.

IV.2.3 Transmisión de símbolos

Tras generar todos los paquetes de codificación éstos se transmiten por el canal. Cada paquete se envía a una dirección y un puerto definidos por la sesión y el canal al que pertenecen. A la hora de mandar se utilizan los bloques “RateControl” y “Scheduler” como muestra la figura 9. El primero de ellos se encarga de enviar los datos a la tasa definida por el canal.

El bloque de “Scheduler” planifica el envío de paquetes. Este bloque permite elegir la frecuencia y el orden en el que se transmiten los paquetes de un determinado objeto. En nuestro caso, realizaremos un estudio para comparar la conveniencia de mandar todos los paquetes de un fichero de forma secuencial y de forma aleatoria.

IV.3. DESCRIPCIÓN DE LIBRERÍAS EN RECEPCIÓN: DECODIFICADOR LDPC

La figura 10 muestra la arquitectura del cliente. Como vemos, comparte algunos elementos con el servidor. En este caso, “jfluteclient” es la librería principal, quien controla las sesiones, canales y su gestión a través de las clases correspondientes.

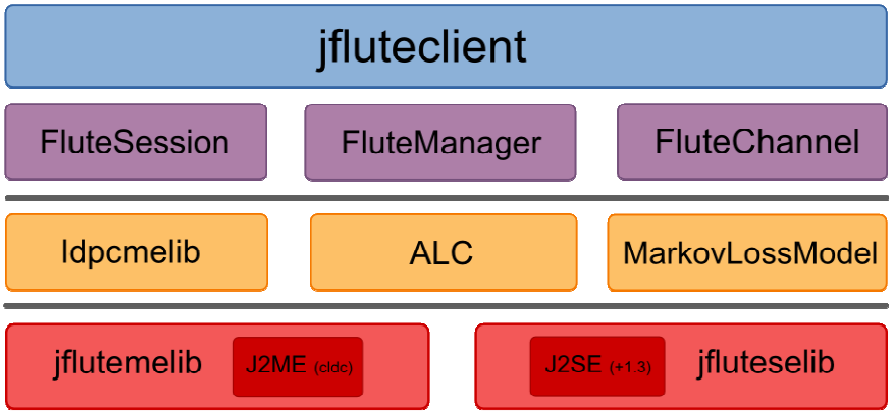


Fig.10. Estructura del cliente de FLUTE.

A un nivel inferior el protocolo ALC es el que gestiona la recepción de los paquetes y se encarga de decodificarlos a través de la librería “ldpcmelib”. Asimismo, para simular las pérdidas en el canal se ha generado una clase que implementa el modelo de pérdidas de Markov.

El cliente está diseñado para J2SE y J2ME, por lo que funciona tanto sobre PC como sobre dispositivos móviles.

IV.3.1 Generación de matriz de paridad

Para la creación de la matriz de paridad se utiliza la misma librería y los mismos métodos empleados en el servidor.

Una vez el receptor ha recibido los parámetros relativos a la codificación (como son la estructura, el n1m3 y la semilla) y ha calculado el número total de símbolos fuente y de símbolos totales de cada uno de los bloques que forman un objeto, puede invocar a la clase “LdpcMatrix” para generar la matriz de paridad, que le permitirá realizar la decodificación de los símbolos recibidos.

IV.3.2 Reconstrucción de símbolos de paridad

La reconstrucción de los símbolos de paridad es el punto clave en el proceso de decodificación. Este paso constituye el punto crítico en lo que respecta a recursos de memoria utilizados en el receptor, lo cual resulta especialmente importante cuando se trabaja con dispositivos móviles.

La clase que contiene los bloques de codificación es “FluteObject”. Se ha implementado una clase llamada “LdpcBlock” que define un bloque de tipo LDPC, y que es quien se encarga de realizar la decodificación, ya que la decodificación se realiza por bloque.

Una vez el cliente haya recibido los parámetros de la codificación (a través de la extensión de cabecera *EXT_FTI*), puede averiguar qué tipo de bloques tendrá que generar. Asimismo, mediante la *EXT_FTI* también obtiene los parámetros que le permiten definir el número de bloques y símbolos que lo componen. Así, cada nuevo paquete que reciba le permitirá asignarlo a un símbolo y un bloque concreto.

Uno de los puntos importantes es el algoritmo de decodificación empleado. En nuestro caso, por simplicidad se ha utilizado el algoritmo de decodificación iterativa. Esto es, la decodificación se basa en la existencia de *buffers* de sumas parciales en cada fila de la matriz de paridad. Así, cuando se recibe un símbolo (ya sea de datos o de paridad) se comprueba en qué filas de la matriz está presente (haciendo uso de la lista enlazada implementada en la librería de LDPC) y se realiza la XOR de los datos que contiene ese paquete con el *buffer* de la fila correspondiente. Hay que tener en cuenta que los símbolos fuente recibidos se guardan en memoria, mientras que los símbolos de paridad no se almacenan directamente, para así no aumentar la memoria utilizada por el cliente.

Cuando se han recibido todos los paquetes que forman parte de la fila excepto uno, los datos del símbolo que faltan se corresponden con la suma parcial del buffer de esa fila, reconstruyendo de esta forma el símbolo que aún no se ha recibido.

El flujograma de la figura 11 representa el proceso que se lleva a cabo cada vez que se recibe un nuevo paquete. Si se recibe un paquete que aún no se tiene y aún no se han recibido todos los

símbolos de un determinado bloque, se obtiene la matriz de paridad asociada a ese bloque, y se suma el símbolo recibido en los *buffers* de las filas en las que participa. Inmediatamente se elimina ese símbolo de la matriz de paridad (la entrada que está a "1" pasa a estar a "0") y se comprueba si se está en disposición de reconstruir algún símbolo, en cuyo caso vuelve a comenzar el proceso.

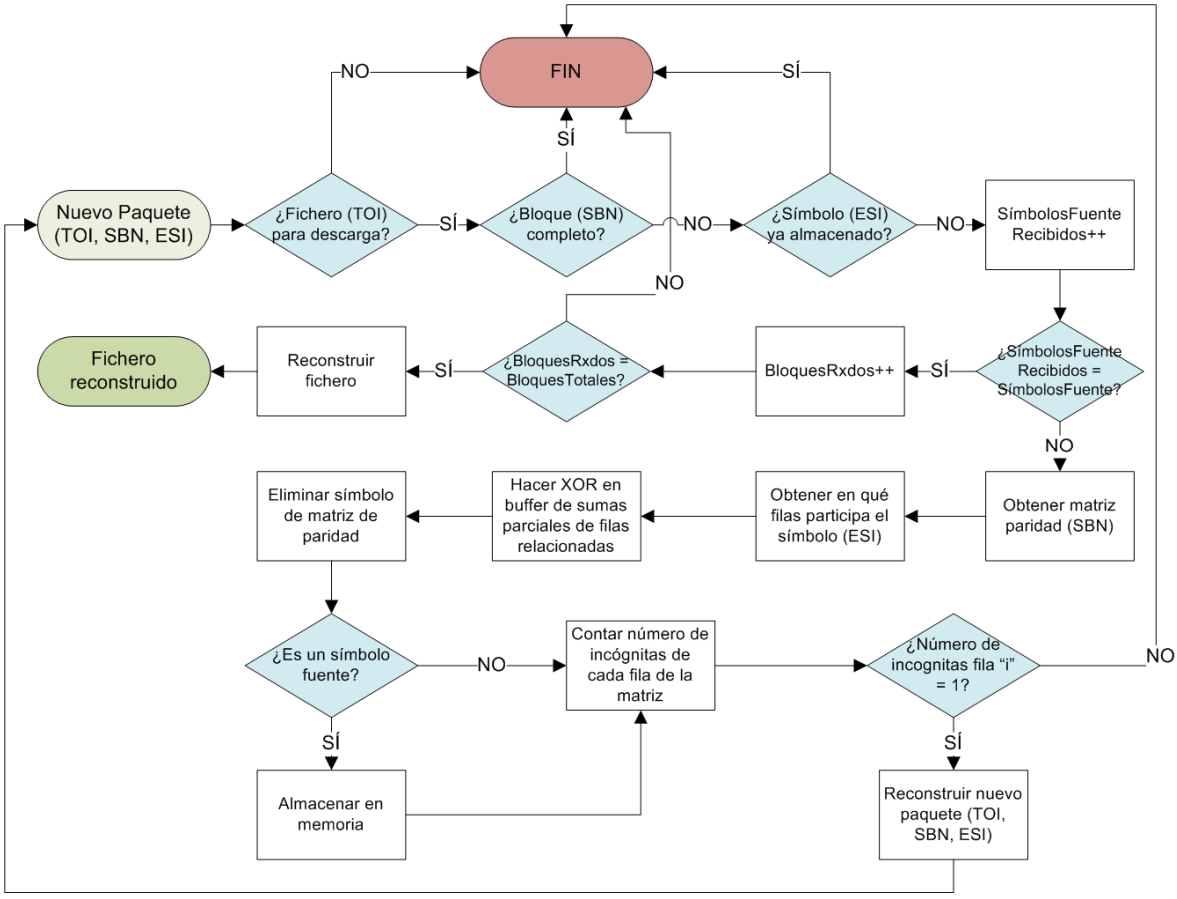


Fig.11. Flujograma de recepción de paquete.

Cada uno de los bloques de un objeto contiene un contador del número total de paquetes fuente que ha recibido. Cuando un bloque ya ha recibido los *k* símbolos fuente, se puede reconstruir ese bloque. Para ello se desencapsulan los paquetes fuente para obtener la carga de cada uno de ellos.

Una vez se han obtenido todos los bloques que forman un fichero se procede a la reconstrucción del mismo. Para ello se crea un fichero con el nombre del objeto (indicado en la FDT del objeto) y con los datos obtenidos de la carga de cada uno de los símbolos de cada bloque.

IV.3.3 Simulador de pérdidas

Para ver el comportamiento que tiene el decodificador cuando se producen pérdidas en el canal se ha implementado un simulador de pérdidas. La razón por la que este simulador se ha desarrollado en el receptor es que en el caso de aplicaciones broadcast, diferentes receptores pueden experimentar distintos modelos de pérdidas, por lo que el comportamiento en recepción es

dependiente del cliente. Un claro ejemplo de esto es una red Wi-Fi, en la que la calidad de la señal depende de la localización del usuario.

Elegir un modelo de pérdidas perfecto para un canal resulta complicado, especialmente en redes inalámbricas, donde es difícil tener en cuenta todos los parámetros (como reflexiones, refracciones, efecto Doppler, interferencias, *fading*…). En nuestro caso vamos a trabajar con un modelo de pérdida de paquetes, que es más independiente de los parámetros del canal (ya que no se trabaja a nivel de bit). Un buen modelo de pérdidas es el modelo de Markov de dos estados (también conocido como modelo de Gilbert), por su simplicidad y por ser un modelo ampliamente utilizado en la literatura [21].

Este modelo se basa en la existencia de dos estados: uno de pérdidas y otro sin pérdidas. La probabilidad de perder o recibir el próximo paquete depende del estado en el que se encuentre. De esta forma se modelan los errores a ráfagas, característicos en las redes inalámbricas, ya que es más probable que se pierda un paquete si el anterior paquete también se ha perdido.

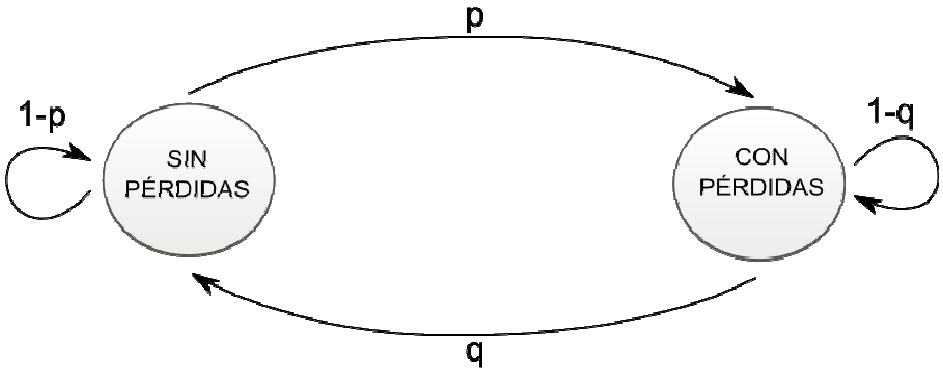


Fig.12. Modelo de Markov de 2 estados.

La probabilidad de pérdida global, así como el tamaño medio de ráfaga vienen definidos por las siguientes ecuaciones:

$$p_{global} = \frac{p}{p+q} \qquad b = \frac{1}{q} \tag{1}$$

Por lo tanto, se pueden modelar las mismas pérdidas con tamaños medios de ráfagas distintos. Por ejemplo, se pueden modelar unas pérdidas medias del 25% con los valores p=0.1 y q=0.3 (longitud media de ráfaga de 3.3 paquetes) y también con p=0.3 y q=0.9 (longitud media de ráfaga de 1.1 paquetes).

V. EVALUACIÓN

En este capítulo se muestran los resultados de la evaluación del códec LDPC explicado en el capítulo anterior. Para ello se han realizado una serie de estudios que reflejan el comportamiento del códec dependiendo de una serie de parámetros, como son las pérdidas, el tamaño del fichero o la tasa de codificación.

El proceso de evaluación consiste en la transmisión, a través del servidor FLUTE, de un fichero de un tamaño determinado, y que ha sido codificado a una cierta tasa con una determinada codificación, y en la correspondiente recepción del fichero a través del cliente FLUTE. Los parámetros que se han medido son:

- Ratio de ineficiencia: es uno de los parámetros de medida de eficiencia más utilizados en la evaluación de mecanismos de reparación de errores. Representa la relación entre el número de paquetes necesarios para decodificar un fichero y el número de paquetes fuente que componen un fichero. Idealmente este valor es 1, lo que significa que cualesquiera de *k* símbolos recibidos (ya sean símbolos fuente o de paridad) son suficientes para reconstruir un fichero:

$$inneficiency\_ratio = \frac{n_{necesarios\_para\_decodificar}}{k} \quad (2)$$

- Número de ciclos: en entornos con pérdidas se necesitan varios ciclos para poder reconstruir un fichero. Cuanto mayor sea el número de ciclos más tiempo costará descargar un fichero.

Las pruebas se han realizado con un Pentium Dual-Core a 2.50 GHz con 2 Gbytes de memoria RAM, utilizando el sistema operativo Windows XP. Asimismo, también se han realizado pruebas en dispositivos móviles, concretamente se ha empleado el Nokia E71, así como el Nokia E90.

El número de simulaciones realizadas varía según el estudio. En las pruebas con dispositivos móviles se han repetido las medidas 30 veces. Por lo que respecta a PC, en los estudios en los que se consideran pérdidas en el canal se han realizado 100 emulaciones, mientras que en el resto de casos se han realizado 200 emulaciones.

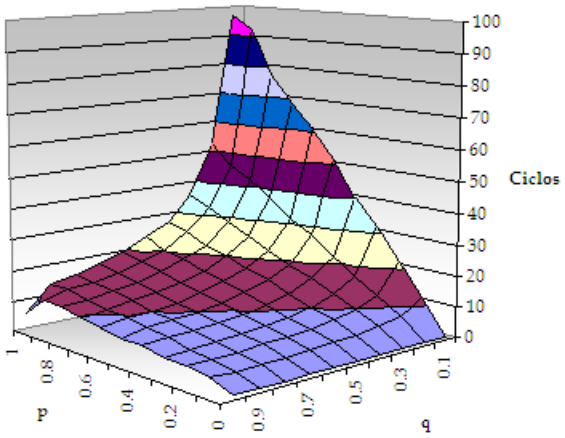
Existen otros estudios en la literatura que evalúan la eficiencia de la codificación LDPC y de las dos variantes que vamos a analizar en esta tesina. De entre las referencias existentes, destacamos los resultados publicados por Vincent Roca y Christophe Neumann, que se pueden encontrar en [22].

V.1. ESTUDIO 1: NÚMERO DE CICLOS PARA RECONSTRUIR UN FICHERO

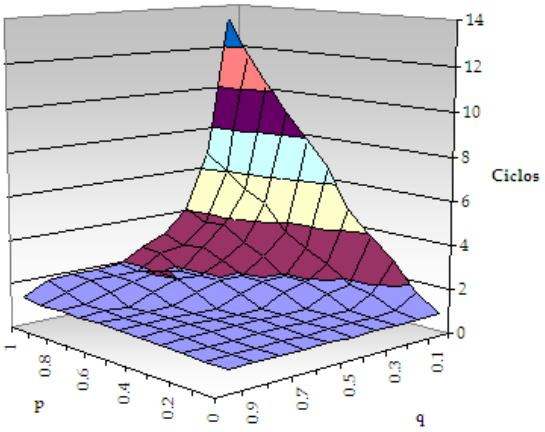
En este primer estudio se muestra el número de ciclos que a un cliente le cuesta reconstruir un fichero en función de las pérdidas del canal, para así comprobar lo conveniente que resulta emplear cualquier mecanismo de codificación. Para ello, se simulan las pérdidas a través de un modelo de Markov de 2 estados, tal como se explicó en el capítulo 4. Se varían los parámetros *p* y *q* para

contemplar todas las pérdidas (aunque los canales con muy altas pérdidas no resultan válidos en la práctica). En las simulaciones no se ha considerado el valor de *q*=0, ya que representa un estado de pérdidas puro. El resto de medidas tiene una precisión decimal (se ha variado cada medida 0.1).

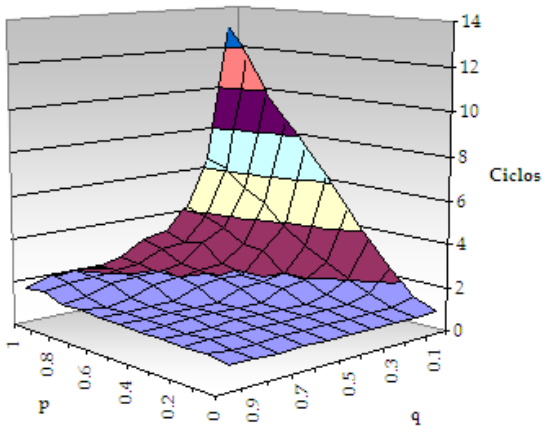
Se ha utilizado un fichero de tamaño 3000 paquetes (unos 4 Mbytes), enviados en un único bloque y se ha empleado una tasa de codificación de 2/3 (es decir, un paquete de paridad por cada dos paquetes de datos). Estos dos parámetros los tomaremos como estándar en el resto de estudios, por lo que, si no se especifica nada distinto, el tamaño del fichero será de 3000 paquetes, el número de bloques será igual a 1 y la tasa de codificación de 2/3. En este estudio en particular, la transmisión de ficheros se ha producido de forma secuencial.



(a) No-FEC



(b) LDPC Staircase



(c) LDPC Triangle

Fig.13. Número de ciclos según codificación.

Las figuras muestran claramente la conveniencia de utilizar codificación, sólo hay que ver la escala de cada una de las gráficas. Cuando las pérdidas son altas el número de ciclos crece de una manera notable. La tendencia del número de ciclos es igual en todos los casos, aumentando de forma exponencial conforme aumentan las pérdidas (aumenta el valor de *p* y disminuye el de *q*).

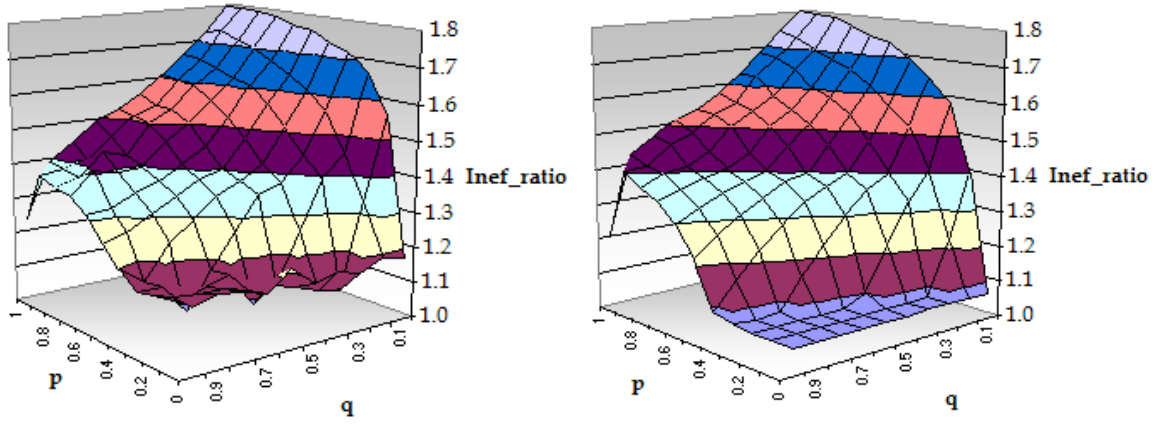
Según estas gráficas, el comportamiento de las dos estructuras de LDPC es similar. En los siguientes estudios profundizaremos y analizaremos el diferente comportamiento que tienen estas dos estructuras según las características de la transmisión.

V.2. ESTUDIO 2: MODELO DE TRANSMISIÓN

A la hora de enviar los paquetes se pueden elegir diferentes estrategias en la transmisión, por ejemplo, enviar todos los paquetes de forma secuencial, o bien enviar primero los paquetes de paridad y luego los paquetes fuente, o transmitirlo todo de forma aleatoria. Ahora bien, ¿afecta el modelo de transmisión a la eficiencia en la decodificación? En este estudio analizamos el comportamiento de dos modelos de transmisión de paquetes, uno secuencial y otro aleatorio, en cuanto a la eficiencia cuando se utiliza codificación LDPC.

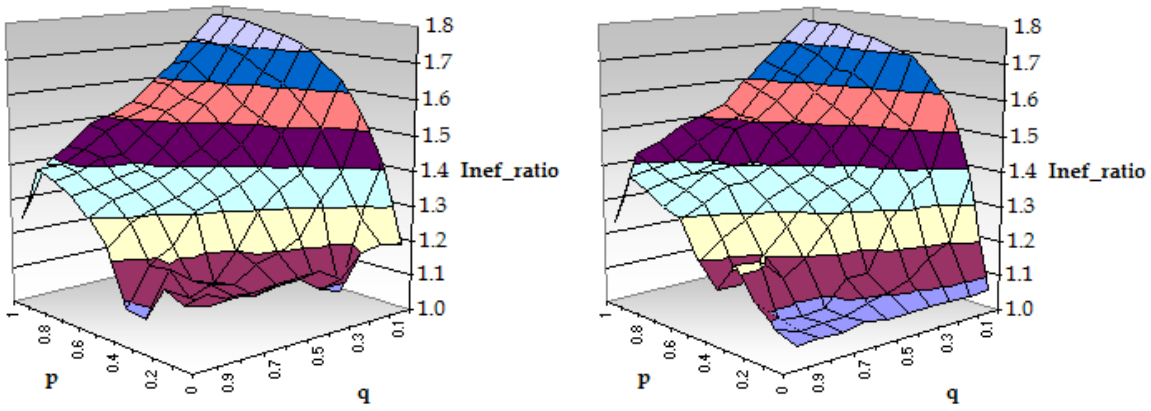
En el modelo secuencial todos los símbolos de un determinado bloque se mandan de forma ordenada, es decir, en primer lugar se envía el símbolo con *Encoding Symbol ID* igual a 0, luego el 1 y así sucesivamente. De esta forma, se transmiten primero los símbolos fuente y luego los símbolos de paridad.

En el modelo aleatorio todos los paquetes de un bloque se mandan aleatoriamente. Es decir, que los símbolos fuente y de paridad van intercalados. Eso sí, cada paquete sólo se envía una vez en cada ciclo, y en cada ciclo se mandan todos los paquetes de un bloque. El parámetro de medida utilizado es el ratio de ineficiencia. El resultado de este estudio se muestra en las siguientes gráficas:



(a) LDPC Staircase, Secuencial

(b) LDPC Staircase, Aleatorio



(c) LDPC Triangle, Secuencial (d) LDPC Triangle, Aleatorio

Fig.14. Modelo de transmisión.

Las gráficas reflejan que en entornos de bajas pérdidas (que se corresponde con los canales donde se suele trabajar), el modelo de transmisión aleatorio presenta un mejor comportamiento y resulta más eficiente que el modelo de transmisión secuencial. Esto es lógico teniendo en cuenta que normalmente las pérdidas se producen a ráfagas y que en la codificación LDPC un símbolo de paridad depende del símbolo anterior, por lo que pérdida de paquetes consecutivos impide la reconstrucción de símbolos fuente. Con altas pérdidas el comportamiento de los dos modelos es similar.

Por lo tanto, vista la conclusión alcanzada, en los siguientes estudios se utilizará el modelo aleatorio en la transmisión de los paquetes.

V.3. ESTUDIO 3: EVALUACIÓN DE LA TASA DE CODIFICACIÓN

La tasa de codificación (o *code rate*), que corresponde con la paridad introducida en los paquetes, es un parámetro fundamental en la transmisión. En este estudio se evalúa la eficiencia de la transmisión en función de la tasa de codificación elegida.

La tasa de codificación corresponde con la relación *k/n*, es decir, que la paridad introducida es de *k-n* paquetes. Otro parámetro que se suele utilizar es el ratio de expansión de FEC (o *FEC ratio*), definido como *n/k*, y que se corresponde con la inversa de la tasa de codificación.

Se han realizado distintas simulaciones variando la tasa de codificación para comprobar la eficiencia de cada tipo de codificación. En este caso, el canal utilizado no tiene pérdidas. Los resultados se pueden ver en la tabla 1. Dicha tabla muestra el ratio de ineficiencia medio, así como la desviación estándar y el intervalo de confianza del 99%, en función de la tasa de codificación y la estructura LDPC.

	STAIRCASE			TRIANGLE		
CR	**MEDIA**	**DESV**	**CONF (99%)**	**MEDIA**	**DESV**	**CONF (99%)**
0.2	1.5118	0.2699	±0.0695	1.9455	0.0333	±0.0086
0.3	1.2037	0.0848	±0.0219	1.4930	0.1643	±0.0423
0.4	1.1549	0.0132	±0.0034	1.1529	0.0294	±0.0076
0.5	1.1145	0.0079	±0.0020	1.1065	0.0101	±0.0018
0.6	1.0885	0.0084	±0.0022	1.0782	0.0076	±0.0014
0.7	1.0624	0.0064	±0.0016	1.0568	0.0052	±0.0009
0.8	1.0414	0.0047	±0.0012	1.0376	0.0039	±0.0007
0.9	1.0214	0.0028	±0.0007	1.0212	0.0034	±0.0006

Tabla 1: Ratio de ineficiencia según la tasa de codificación

Los valores de la desviación estándar y el intervalo de confianza son bastante buenos, lo que nos asegura que el número de muestras tomadas es correcto. Los resultados de la tabla se reflejan en la siguiente gráfica (se ha ampliado el eje de valores de *code rate* superiores a 0.4 para ver en detalle el comportamiento de cada una de las estructuras):

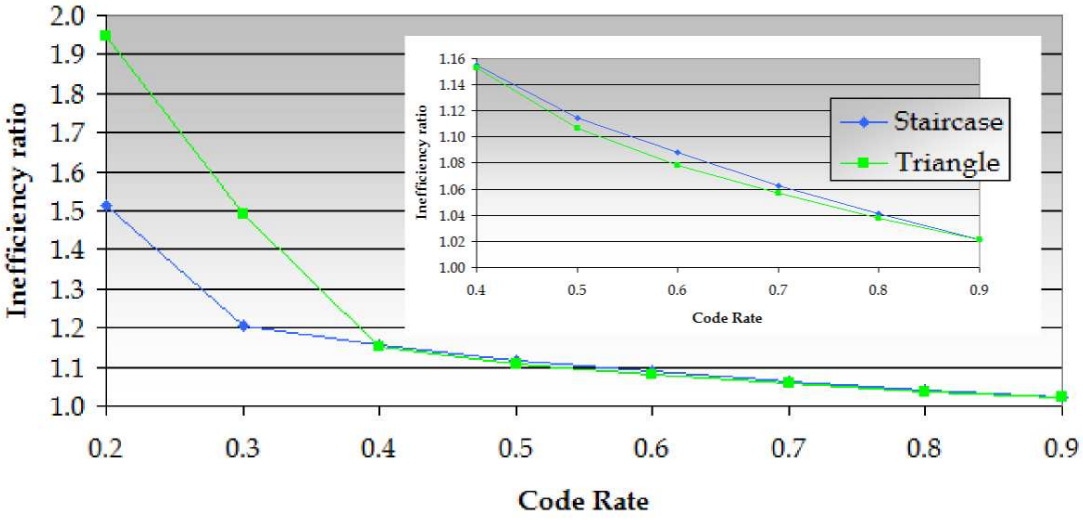


Fig.15. Evaluación de la tasa de codificación.

Conforme aumenta la tasa de codificación el ratio de ineficiencia es menor (y por lo tanto, mejor). La figura 15 refleja que la estructura LDPC Staircase resulta más eficiente cuando la tasa de codificación es inferior a 0.4, mientras que LDPC Triangle ofrece mejores resultados para tasas de codificación superiores a dicho valor. La figura 16 muestra los mismos resultados en función del ratio de expansión de FEC (la inversa de la tasa de codificación), para poder comparar de esta forma con los resultados publicados en [22].

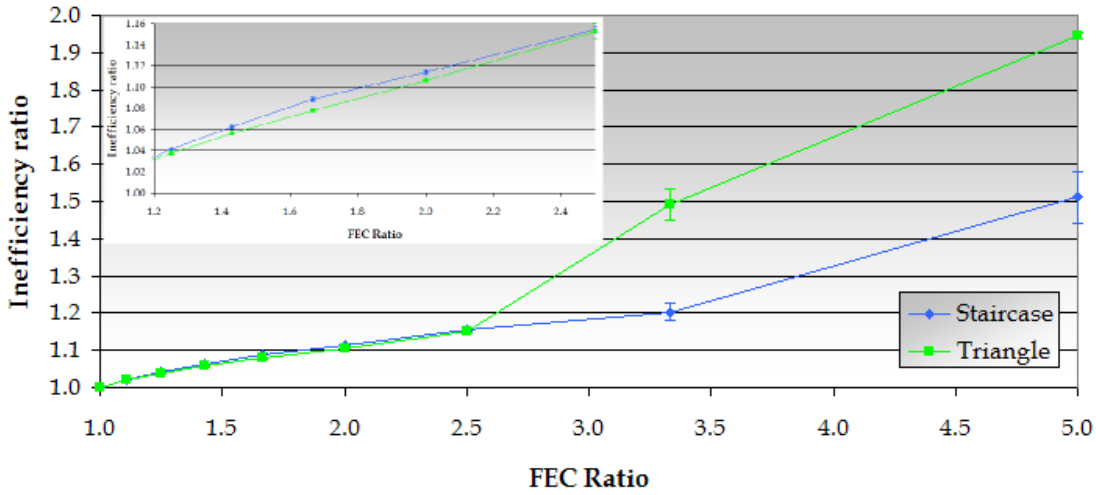


Fig.16. Evaluación del ratio de expansión FEC.

Cuanto mayor es el *code rate* (y en consecuencia menor es el *FEC ratio*), menos paquetes de paridad se envían, por lo que, en entornos sin pérdidas el ratio de ineficiencia será más bajo (ya que se reciben menos paquetes "inútiles"). Idealmente, en un entorno sin pérdidas si la tasa de codificación es 1 (es decir, no se codifica) la tasa de ineficiencia es 1. Pero, desgraciadamente, todos los canales tienen pérdidas. Para ver el comportamiento en un entorno con pérdidas, se ha realizado el mismo estudio en un canal con unas pérdidas elevadas (del 25%), con valores p=0.1 y q=0.3, que se refleja en la siguiente gráfica.

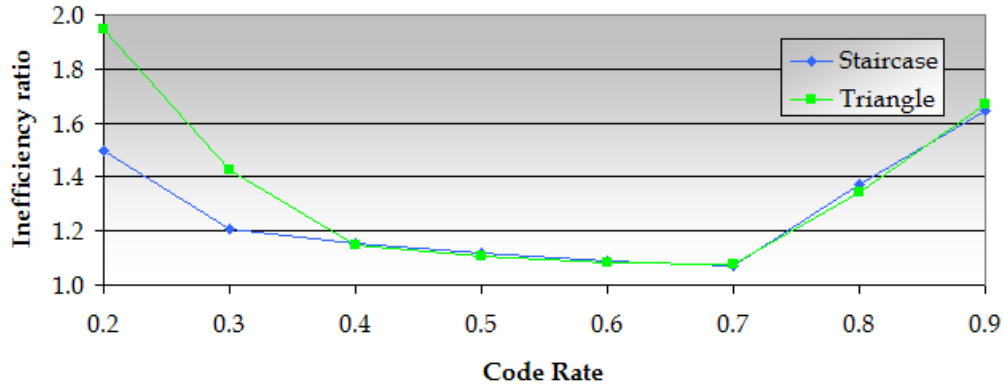


Fig.17. Evaluación de la tasa de codificación con pérdidas (p=0.1, q=0.3).

Con esta gráfica se concluye que para elegir una tasa de codificación adecuada hay que tener en cuenta las pérdidas del canal. En este caso, para un canal con esas características el valor de *code rate* igual a 0.7 es el que ofrece mejores resultados.

V.4. ESTUDIO 4: EVALUACIÓN DEL TAMAÑO DEL FICHERO

El ratio de ineficiencia también depende del tamaño del contenido que se está enviando. Y para comprobarlo, vamos a realizar un estudio del comportamiento del códec según el tamaño del fichero. Para ello, fijando una tasa de codificación de 2/3 y utilizando un solo bloque, enviamos

varios ficheros con distinto tamaño a través de un canal sin pérdidas. La tabla 2 muestra los resultados obtenidos, donde el tamaño corresponde con el número de paquetes (o símbolos) que componen un bloque (se puede hallar el tamaño del fichero teniendo en cuenta que cada paquete ocupa 1428 bytes).

	STAIRCASE			TRIANGLE		
Tamaño	**MEDIA**	**DESV**	**CONF (99%)**	**MEDIA**	**DESV**	**CONF**
10	1.1736	0.1227	±0.0224	1.2099	0.0490	±0.0126
50	1.1210	0.0805	±0.0147	1.1511	0.0554	±0.0143
200	1.0905	0.0204	±0.0053	1.1038	0.0218	±0.0056
1000	1.0777	0.0110	±0.0028	1.0736	0.0086	±0.0022
5000	1.0706	0.0041	±0.0011	1.0612	0.0048	±0.0012
10000	1.0681	0.0029	±0.0008	1.0593	0.0072	±0.0019

Tabla 2: Ratio de ineficiencia según el tamaño del fichero

Como vemos en la tabla, con ficheros grandes los resultados del ratio de ineficiencia son mucho mejores. Por ejemplo, para ficheros de 10000 paquetes (unos 14 Mbytes), para el caso de Triangle el ratio de ineficiencia es de 1.0593. Esto significa que sólo hará falta recibir un 5.93% más de paquetes que componen un fichero para poder reconstruirlo. Esto implica que se está proporcionando fiabilidad en la transmisión sin aumentar excesivamente el tiempo de reconstrucción en recepción.

Los resultados se muestran, a escala logarítmica, en la siguiente figura:

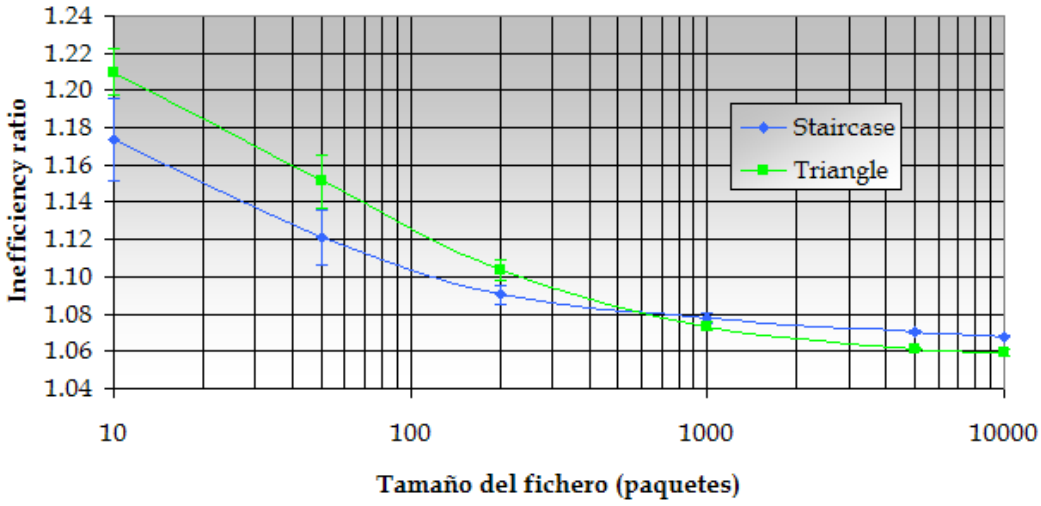


Fig.18. Evaluación del tamaño del fichero.

Cuanto mayor es el fichero el ratio de ineficiencia es menor, por lo que la codificación LDPC es más eficiente con ficheros grandes. Esto es consecuencia de la generación de la matriz de paridad:

cuantas más entradas tiene la matriz, el algoritmo de generación de la matriz hace que los símbolos estén relacionados de una forma más óptima.

Por lo que respecta al tipo de estructura, la tendencia de la gráfica demuestra que con ficheros pequeños y a una tasa de codificación de 2/3 la estructura Staircase resulta más eficiente, mientras que para ficheros grandes LDPC Triangle ofrece mejores resultados.

Otro estudio que surge a raíz de esta conclusión es el número de bloques que se utiliza para transmitir. Resulta lógico pensar que si el códec es más eficiente cuantos más paquetes componen el bloque, será mucho más eficiente transmitir el fichero en un solo bloque que en varios bloques, ya que cada uno de ellos tendrá menor número de símbolos.

Al respecto, la siguiente gráfica muestra el ratio de ineficiencia de un fichero formado por 6400 paquetes, que se ha transmitido utilizando distinto número de bloques:

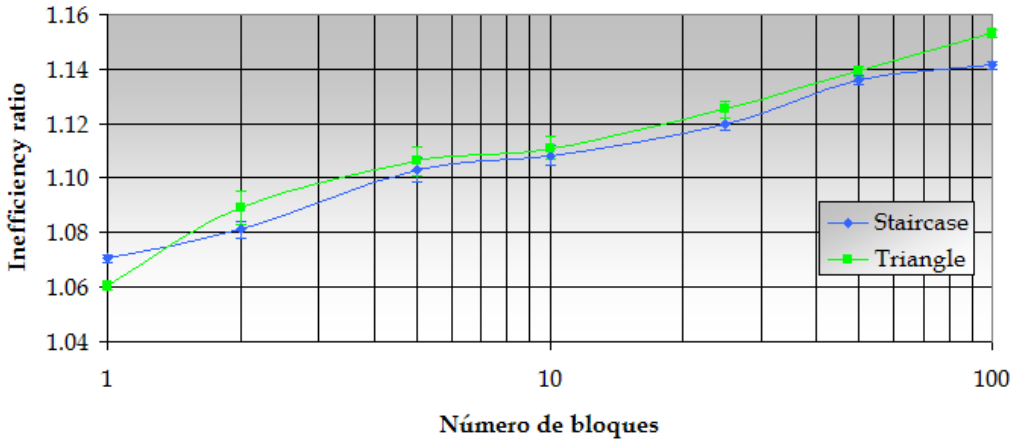


Fig.19. Evaluación del número de bloques.

Las conclusiones alcanzadas en este caso son consecuencia de las anteriores. Cuantos más bloques haya más ineficiente es la codificación. Cuanto mayor es el número de bloques, menos símbolos hay por bloque, por lo que la estructura Staircase resulta más eficiente que la Triangle.

Aunque resulta, pues, más eficiente emplear un solo bloque para la transmisión, esto no significa que siempre sea lo más conveniente. Dependiendo de la situación, puede ser preferible enviar un fichero en varios bloques, debido, sobre todo, a limitaciones en cuanto a la memoria. Hay que tener en cuenta que cuanto mayor sea el bloque más consumo de memoria se requiere. Este es uno de los aspectos que queda fuera del alcance de esta tesina, y que se contempla como trabajo futuro.

V.5. ESTUDIO 5: EVALUACIÓN DEL NÚMERO DE 1s POR COLUMNA

En la generación de la matriz de paridad (concretamente en la submatriz izquierda), cada símbolo fuente puede formar parte de un número determinado de ecuaciones (denominado como N1). Este número es fijo para cada matriz y suele ser igual a 3, tal como lo recomienda la RFC de LDPC [1].

En este estudio, veremos cómo afecta dicho valor a la eficiencia de la codificación. La figura 20 muestra el ratio de ineficiencia obtenido cuando se varía N1 entre 3 y 8.

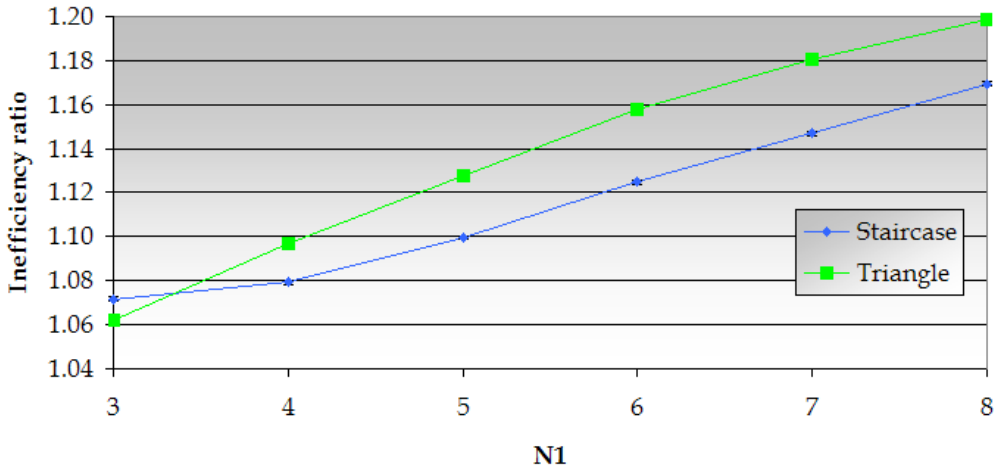


Fig.20. Evaluación del número de 1s por columna en la matriz de paridad.

Los resultados muestran que no resulta eficiente aumentar el valor de N1. Valores inferiores a 3 no resultan válidos. Además, el que haya más entradas en la matriz de paridad implica mayor consumo de memoria y un mayor procesamiento cuando se recibe cada paquete.

No obstante, hay que indicar que el estudio depende del algoritmo de decodificación utilizado. En nuestro caso, se ha empleado el algoritmo de decodificación iterativa por su simplicidad y por su bajo consumo de memoria. Diversos estudios, como el que se puede encontrar en [23], señalan que emplear otro algoritmo de decodificación (por ejemplo el esquema basado en la eliminación Gaussiana) permite reducir el ratio de ineficiencia. En dicho estudio se refleja que, utilizando un esquema de decodificación basado en la eliminación Gaussiana, el aumento de N1 supone una mejora del ratio de ineficiencia para LDPC Staircase pero, a su vez, hace aumentar los recursos de memoria.

Por lo tanto, podemos concluir este estudio diciendo que, utilizando el algoritmo de decodificación iterativa, aumentar el valor de N1 no provoca una mejora en el ratio de ineficiencia, por lo que el valor óptimo resulta *N1=3*.

V.6. ESTUDIO 6: EVALUACIÓN EN DISPOSITIVOS MÓVILES

El último estudio que se ha realizado consiste en la evaluación del códec implementado en dispositivos móviles y en un entorno inalámbrico. Concretamente, los terminales utilizados han sido el Nokia E71 y Nokia E90. La red inalámbrica escogida ha sido Wi-Fi.

El procedimiento es similar al de los estudios anteriores. Un servidor envía por Wi-Fi un fichero a una dirección y un puerto determinados y un receptor se descarga dicho fichero a través de un teléfono móvil. Para ello, en el dispositivo móvil se ha instalado una aplicación cliente receptora de ficheros FLUTE, que se conecta a una sesión y un canal para descargarse un contenido, todo ello a

través de Wi-Fi. Cuando se ha descargado, muestra por pantalla el valor del parámetro que se desea medir (en nuestro caso el ratio de ineficiencia).

En este caso, para contrastar las conclusiones alcanzadas en los estudios anteriores en un entorno real, se ha optado por realizar tres pruebas en las que se varía la tasa de codificación y el tamaño del fichero, así como la estructura LDPC empleada. Por las conclusiones alcanzadas anteriormente, el modelo de transmisión es aleatorio, el número de bloques es igual a 1 y el número de 1s en la submatriz izquierda es igual a 3.

La tabla 3 muestra los resultados obtenidos:

		STAIRCASE			**TRIANGLE**		
CR	**Tamaño**	**MEDIA**	**DESV**	**CONF 99%**	**MEDIA**	**DESV**	**CONF 99%**
0.20	300	1.4916	0.0420	±0.0216	2.1419	0.0283	±0.0146
0.67	300	1.4584	0.0733	±0.0378	1.4527	0.0070	±0.0036
0.67	1500	1.4187	0.0168	±0.0087	1.4294	0.0229	±0.0118

Tabla 3: Evaluación en entorno inalámbrico

La primera conclusión al realizar las pruebas ha sido que la red Wi-Fi es muy variante e inestable. Y aunque los datos de la desviación estándar y el intervalo de confianza son relativamente buenos, eso no implica que las medidas resulten definitivas. Las pérdidas en el canal dependen en gran parte del número de dispositivos inalámbricos cercanos, ya que aumenta el número de colisiones en la red. La magnitud del tiempo es un parámetro importante al respecto (dependiendo de la hora del día, hay más o menos nodos interferentes en el canal). Es por ello que se ha intentado realizar las medidas a las mismas horas para tener una referencia común.

Por lo que respecta a la codificación, en primer lugar vemos que las pérdidas en el canal afectan mucho a la eficiencia. Mientras que en el resto de casos obteníamos normalmente valores inferiores a 1.1 en cuanto al ratio de ineficiencia, en pruebas con Wi-Fi los valores obtenidos han sido siempre superiores a 1.25.

Algunas de las conclusiones alcanzadas en estudios anteriores también se reflejan en la tabla. Por ejemplo, con tasas de codificación bajas la estructura Staircase mejora considerablemente a la Triangle, reduciéndose esa diferencia conforme la tasa de codificación aumenta. Utilizar una tasa de codificación mayor supone una disminución del ratio de ineficiencia. Por otro lado, un aumento del tamaño del fichero conlleva una mejora del ratio de ineficiencia.

VI. CONCLUSIONES Y TRABAJO FUTURO

VI.1. CONCLUSIONES

A lo largo de esta tesina las ventajas de utilizar codificación a nivel de paquete han quedado patentes. En todos los entornos se producen pérdidas, ya sea en mayor o menor medida, y la utilización de mecanismos de corrección de errores resulta necesaria.

La codificación LDPC es un claro ejemplo de mecanismo FEC. Su utilización permite reducir de manera considerable el número de ciclos necesarios para reconstruir un fichero, y por lo tanto el tiempo de descarga. La reducción en dicho tiempo es aún mayor en canales con pérdidas elevadas.

Por otro lado, la planificación en el envío de paquetes es un parámetro que afecta a la eficiencia de la transmisión. Un modelo de envío aleatorio resulta mucho más eficiente que uno secuencial, ya que es más inmune a las pérdidas de paquetes en forma de ráfagas.

En lo que respecta a las variantes de LDPC, los distintos estudios nos han permitido comparar los dos tipos de estructuras y verificar cuál es mejor dependiendo de las circunstancias. Así, cuando se emplean tasas de codificación inferiores a 0.4 (que representa una paridad superior al 60%) resulta más eficiente la utilización de LDPC Staircase, mientras que LDPC Triangle se comporta mejor con tasas superiores.

Asimismo, la eficiencia de la codificación depende del tamaño del contenido que se envía. Cuanto más grande sea un fichero, más eficiente es la codificación. Esto provoca que resulte más óptima, en términos de eficiencia, la transmisión utilizando un único bloque. Para ficheros pequeños, la estructura Staircase ofrece mejores resultados.

La elección, pues, de un tipo de codificación u otra depende de muchos aspectos. Al margen de estas conclusiones con respecto a la tasa de codificación y al tamaño, es necesario considerar otros factores. Por un lado, hay que evaluar el canal de transmisión para, en función de las pérdidas, decidir qué codificación y qué tasa emplear. Un claro ejemplo lo hemos visto en el estudio en el canal Wi-Fi. Los ficheros a enviar resultan otro factor clave. El tamaño, así como la relevancia del contenido decidirán qué parámetros de codificación utilizar (un fichero más importante requerirá de más protección).

Por otra parte, se debe tener en cuenta a quién va dirigido el contenido. Las capacidades de memoria y de procesamiento de los clientes pueden condicionar las características de la codificación. Los dispositivos móviles, por ejemplo, pueden tener problemas al decodificar un fichero muy grande. Por lo tanto, sería conveniente dividir el fichero en varios bloques y utilizar una estructura y un algoritmo de decodificación que minimice el procesamiento requerido para realizar la decodificación.

VI.2. TRABAJO FUTURO

El trabajo futuro que se desprende de esta tesina se puede estructurar en cuatro grandes bloques: estudio en redes inalámbricas, estudio del consumo, estudio de otras codificaciones y estudio de la transmisión.

En esta tesina se han realizado una serie de pruebas utilizando Wi-Fi como red inalámbrica multicast. Como hemos visto, esta red es muy dependiente del entorno y, por lo tanto, muy sensible a errores en el canal. En este sentido, una línea que se deja abierta es la realización de un estudio más exhaustivo que tenga en cuenta la carga de la red y el número de redes y nodos interferentes.

Asimismo, tal como se explicó en el segundo capítulo, existen otro tipo de redes inalámbricas, entre ellas DVB-H. Puesto que se dispone de un transmisor y de un receptor DVB-H, uno de los objetivos inmediatos es la evaluación del códec implementado en este tipo de redes.

Al respecto, es necesario analizar el canal donde se realiza la transmisión para elegir unos parámetros de codificación u otros, tal como vimos en el estudio 3. En función de las pérdidas del canal resulta más conveniente transmitir a una tasa de codificación o a otra. Por ello, hay que estudiar el modo en el que los receptores pueden informar al transmisor sobre las pérdidas que se están detectando en el canal, para que el transmisor envíe lo más óptimamente posible según las condiciones del entorno.

Uno de los puntos que hemos dejado pendientes en la tesina es el tema de la memoria. El consumo en el procesamiento del códec en transmisión, pero principalmente en recepción, es un punto crítico, y más si se trabaja con dispositivos portátiles. Existen varios estudios al respecto, como el que se puede encontrar en [23], que evalúan la eficiencia de memoria de ambos tipos de estructuras LDPC.

Precisamente, una de las principales dificultades que nos hemos encontrado a la hora de evaluar el códec han sido problemas de memoria cuando transmitíamos ficheros grandes. Esto ha resultado especialmente crítico en las pruebas con móviles. Por lo tanto, uno de nuestros principales objetivos es estudiar los recursos consumidos en los dispositivos e intentar reducir la carga. El algoritmo de decodificación utilizado es un punto crítico en este aspecto. Por simplicidad se ha empleado el algoritmo de decodificación iterativa, pero existen otro tipo de algoritmos que reducen los recursos en el proceso de decodificación: el esquema de decodificación iterativa de Zyablov (ID), la eliminación Gaussiana (GE) y un esquema híbrido ID/GE [24]. Un estudio de estas alternativas permitirá mejorar las características de nuestro códec.

Por otro lado, más allá de la codificación LDPC, se pretende estudiar la codificación Raptor, debido a los buenos resultados que estos códigos ofrecen (con ratios de ineficiencias cercanos a 1). Para ello, se analizará la RFC de Raptor, así como las futuras variantes de esta codificación. Entre ellas encontramos los códigos Raptor-Q, una codificación en fase de experimentación (el *draft* 02

se publicó en marzo de 2010) [25] que promete ofrecer mejores resultados que la codificación original.

También se analizarán las distintas variantes de la codificación LDPC que van surgiendo, entre ellas señalar los códigos HLDPC (*Hyper LDPC*) [26], una evolución de los LDPC que ofrecen una menor complejidad.

En último lugar, la implementación de mecanismos de FEC en la librería FLUTE permite aumentar las funcionalidades de nuestra implementación en cuanto a la planificación de paquetes. Una de las aplicaciones principales puede ser proporcionar diferente protección frente a errores. Así, en un carrusel de envío de contenido, en función de la importancia o de la popularidad del fichero a enviar se le puede ofrecer mayor o menor protección. Recordemos que la codificación va por fichero, por lo que un fichero muy popular puede tener mayor paridad que otro fichero con menor popularidad. Incluso, más allá de la redundancia, se puede utilizar un tipo de codificación u otra en función de las características del fichero, teniendo en cuenta las conclusiones alcanzadas en los estudios realizados en esta tesina.

AGRADECIMIENTOS

En primer lugar, me gustaría dar las gracias a mis compañeros Román Belda y Francisco Fraile, por mostrarme el camino y caminarlo conmigo. Gracias por su impagable ayuda, por su paciencia y por su interés en la tesina. Dos modelos a seguir.

También querría expresar mi gratitud a Pau Arce, y agradecer la ayuda proporcionada por mi director de tesina Juan Carlos Guerri, así como por el resto de compañeros del COMM: Tito, Víctor, Patri y Wilder. Gracias por el ambiente que creáis. Un placer trabajar con vosotros.

Mi más sincero agradecimiento a mis padres y a mi hermana, por todo lo que representan. Una referencia para mí.

Por último, debo indicar que este trabajo de investigación ha sido desarrollado en el Grupo de Comunicaciones Multimedia (COMM) de iTEAM, y que se enmarca en el proyecto “Redes Híbridas” (TSI-020302-2008-94), financiado por el Ministerio de Industria, Turismo y Comercio.

BIBLIOGRAFÍA

[1] V. Roca, C. Neumann, and D. Furodet, *Low Density Parity Check (LDPC) Staircase and Triangle Forward Error Correction (FEC) Schemes*. IETF RFC 5170, June 2008.

[2] T. Paila, M. Luby, R. Lehtonen, V. Roca and R. Walsh, *FLUTE – File Delivery Over Unidirectional Transport*. IETF RFC 3926, October 2004.

[3] G. Faria, J. Henriksson, E. Stare and P. Talmola, *DVB-H: Digital Broadcast Services to Handheld Devices*. Proc. of the IEEE, vol. 94, no. 1, pp. 194-209, January 2006.

[4] *Mobile Broadcast/Multicast Service (MBMS)*. White Paper, August 2004.

[5] *Integrated Mobile Broadcast (IMB): The Power of Predictive Broadcasting for 3G Multimedia Applications*. White Paper, September 2009.

[6] S. Park and S. Jeong, *Mobile IPTV: Approaches, Challenges, Standards, and QoS Support*. IEEE Internet Computing, vol. 13, no. 3, pp. 23-31, June 2009.

[7] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo and J. Crowcroft, *Asynchronous Layered Coding (ALC) Protocol Instantiation*. IEFT RFC 3450, December 2002.

[8] M. Handley and V. Jacobson, *SDP: Session Description Protocol*. IEFT RFC 2327, April 1998.

[9] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, L. Handley and J. Crowcroft, *Layered Coding Transport (LCT) Building Block*. IEFT RFC 3451, December 2002.

[10] M. Watson, M. Luby and L. Vicisano, *Forward Error Correction (FEC) Building Block*. IETF RFC 5052, August 2007.

[11] M. Watson, *Basic Forward Error Correction (FEC) Schemes.* IETF RFC 5445, March 2009.

[12] I. S. Reed and G. Solomon, *Polynomial Codes over Certain Finite Fields*. SIAM Journal of Applied Math, vol. 8, pp. 300-304, 1960.

[13] J. Lacan, V. Roca, J. Peltotalo and S. Peltotalo, *Reed-Solomon Forward Error Correction (FEC) Schemes.* IETF RFC 5510, April 2009.

[14] A. Shokrollahi, *Raptor Codes*. IEEE Transactions on Information Theory no. 6, June 2006.

[15] M. Luby, *LT Codes*. Proc. IEEE Symposium on Foundations of Computer Science (FOCS), Vancouver, Canada, 2002.

[16] P. Cataldi, M. Pedròs, M. Grangetto and E. Magli, *Implementation and performance evaluation of LT and Raptor codes for multimedia applications*. Proc. of the 2006 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'06), California, USA, 2006.

[17] M. Luby, A. Shokrollahi, M. Watson and T. Stockhammer, *Raptor Forward Error Correction Scheme for Object Delivery*. IETF RFC 5053, September 2007.

[18] R. G. Gallager, *Low Density Parity Check Codes*. IEEE Transactions on Information Theory, 8(1), January 1962.

[19] D. MacKay and R. Neal, *Good codes based on very sparse matrices.* In 5th IAM Conference: Cryptography and Coding, LNCS No. 1025, 1995.

[20] S. Park and K. Miller, *Random Number Generators: Good Ones are Hard to Find*. Communications of the ACM, Vol. 33, No. 1, pp. 87-88, January 1990.

[21] H. Bai and M. Atiquzzaman, *Error modeling schemes for fading channels in wireless communications: a survey*. IEEE Communications Surveys and Tutorials, 5(2), 2003.

[22] V. Roca and C. Neumann, *Design, Evaluation and Comparison of Four Large Block FEC Codecs, LDPC, LDGM, LDGM Staircase and LDGM Triangle, plus a Reed-Solomon Small Block FEC Codec*. INRIA Research Report RR-5225, June 2004.

[23] M. Cunche and V. Roca, *Optimizing the Error Recovery Capabilities of LDPC-Staircase Codes Featuring a Gaussian Elimination Decoding Scheme*. Proc. of the 10th IEEE International Workshop on Signal Processing for Space Communications (SPSC), Rhodes Island, Greece, October 2008.

[24] M. Cunche and. V. Roca, *Improving the Decoding of LDPC Codes for the Packet Erasure Channel with a Hybrid Zyablov Iterative Decoding/Gaussian Elimination Scheme*. INRIA Research Report RR-6473, March 2008.

[25] M. Luby, A. Shokrollahi, M. Watson and T. Stockhammer, *RaptorQ Forward Error Correction Scheme for Object Delivery*. IETF Draft 02, March 2010.

[26] K. Nybom and J. Björkqvist, *HLDPC Codes – Low Density, Low Complexity, Efficient Erasure Correcting Codes*. Proc. of the 13th European Wireless Conference, Paris, France, April 2007.

GLOSARIO

El siguiente listado muestra las siglas utilizadas en la tesina:

3G	3rd Generation
3GPP	3rd Generation Partnership Project
ADSL	Asymmetric Digital Subscriber Line
ALC	Asynchronous Layered Coding
ARQ	Automatic Repeat Request
CD	Compact Disc
DVB	Digital Video Broadcasting
DVB-C	Digital Video Broadcasting – Cable
DVB-H	Digital Video Broadcasting – Handheld
DVB-S	Digital Video Broadcasting by Satellite
DVB-T	Digital Video Broadcasting – Terrestrial
DVD	Digital Versatile Disc
ESG	Electronic Service Guide
ESI	Encoding Symbol Identifier
FDT	File Delivery Table
FEC	Forward Error Correction
FLUTE	File Delivery over Unidirectional Transport
FTI	FEC Object Transmission Information
HLDPC	Hyper Low Density Parity Check
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IMB	Integrated Mobile Broadcast
IP	Internet Protocol
IPTV	Internet Protocol Television
J2ME	Java 2 Micro Edition
J2SE	Java 2 Standard Edition
LCT	Layered Coding Transport
LPDC	Low Density Parity Check
LT	Luby Transform
MBMS	Multimedia Broadcast and Multicast Services
MDS	Maximum Distance Separable
MIME	Multipurpose Internet Mail Extensions
PC	Personal Computer
RAM	Random Access Memory
Raptor	Rapid Tornado

RFC	Request for Comments
SAP	Session Announcement Protocol
SBN	Source Block Number
SDP	Session Description Protocol
TCP	Transmission Control Protocol
TOI	Transport Object Identifier
TSI	Transport Session Identifier
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications Systems
URI	Uniform Resource Identifier
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
XML	Extensible Markup Language

ANEXOS

A continuación se adjuntan los artículos en los que el autor de la tesina ha participado, éste es el listado:

- CONGRESOS
  - R. Belda, I. de Fez, F. Fraile, V. Murcia, P. Arce and J. C. Guerri, *Multimedia System for Emergency Services over TETRA-DVBT Networks*. Proc. 34th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA), Parma, Italy, September 2008.
  - F. Fraile, I. de Fez and J. C. Guerri, *Modela-TV: Service Personalization and Business Model Management for Mobile TV*. In 7th European Interactive TV Conference, Leuven, Belgium, June 2009.
  - A. Gil, F. Fraile, M. Ramos, I. de Fez and J. C. Guerri, *Personalized Multimedia Touristic Services for hybrid broadcast/broadband mobile receivers*. International Conference on Consumer Electronics (ICCE), Las Vegas, USA, January 2010.
- REVISTAS
  - A. Gil, F. Fraile, M. Ramos, I. de Fez and J. C. Guerri, *Personalized Multimedia Touristic Services for hybrid broadcast/broadband mobile receivers*. IEEE Transactions on Consumer Electronics, 2010 (aceptado para publicación).

Multimedia System for Emergency Services over TETRA-DVBT Networks

Román Belda, Ismael de Fez, Francisco Fraile, Víctor Murcia, Pau Arce, Juan Carlos Guerri
Institute of Telecommunication and Multimedia Applications
Valencia, Spain
E-mail: jcguerri@dcom.upv.es

Abstract

This paper describes the main challenges of implementing a Multimedia on Demand system (TetraMoD) for emergency scenarios. The objective of this system is to offer a complete solution for providing multimedia services (Video on Demand and Reliability File Transfer) to a rescue team in an emergency situation. This multimedia system will improve the traditional emergency communications. We propose to integrate the use of different networks, TETRA and DVBT, characterized by a wide geographical coverage and broadband broadcast, respectively. This system consists of the following elements: a TetraMoD Gateway - the unit deployed on as interface between TETRA and DVBT networks- and a Middleware TetraMoD Client -the software for the client terminal. A real scenario and example of communication using standard protocols (SDP, RTP/RTCP, RTSP, FLUTE) are showed.

1. Introduction

The possibility of accessing to up-to-date information, in real time and multiple formats (audio, video, images, text, binary files, etc) represents a key factor for the majority of communication systems. Specially important are emergency environments, disaster situations, medical attention, security and so on; where the communication between emergency teams such as police, fire-fighters, ambulances and the control headquarters or hospitals are crucial for making the right decisions at the right time.

Nowadays, most telecommunication systems targeting security and emergency response are based on the Terrestrial Trunked Radio (TETRA) network [1]. This standard is deployed worldwide, especially on Europe, and offers both support for voice communications and as a data channel for low speed transmissions (28.8 kbps per carrier). Nevertheless, despite its evolution (Release 2) so that it supports new data services, it won't be able to carry multimedia services such as real-time video, image downloading, file transfer, remote patient monitoring and so forth needed in next generation public safety systems.

Multiple solutions have been proposed based on satellite, GMTS/UMTS, WiMAX and other technologies in order to solve the problem consisting of sending multimedia information to emergency teams, some of them focused on ad hoc networks deployment in an emergency scenario. However, plus the diffusion capability, the adopted solution should take into account some other equally important aspects like coverage, scalability, security, QoS support, and low implementation cost.

Amongst the different works related to emergency environments and developed over the last years, we can mention the Wideband Wireless Local Area Network (WWLAN) Project [2], focused on the evaluation by simulation of heterogeneous satellite and WLAN network facilities, allowing audio-video communications between the fire brigade acting in a disaster area and the Control Centre that coordinates operations. Another project would be ADAMO [3], which is focused on how an ad-hoc wireless network can be securely connected and extended into a larger wireless network based upon technologies such as WIMAX or WIFI Mesh. The ADAMO project also investigates the integration of this environment with a TETRA network used by emergency services. Another work is WISECOM Project (Wireless Infrastructure over Satellite for Emergency COMmunications project) [4] focused on a description of the unit deployed on the disaster area. It is used as an interface between the selected satellite system and different wireless local access technologies (GSM, UMTS and TETRA over satellite).

However, there is not any solution based on integration between TETRA and DVB-T networks. DVB-T network [5] should be considered as an excellent candidate for broadcasting in view of features like wide coverage, multicast transmission capability, bandwidth, multimedia transmission, and low cost DVB-T receiver availability. The proposed solution in this paper is based on TETRA and DVB-T networks integration, which allows multimedia communications to improve the traditional emergency communications.

The rest of the paper is organized as follows: in Section II the considered scenario and protocols are described; in Section III a description of the middleware software deployed in the client terminal and the gateway implemented as interface between TETRA and multimedia server is provided. Next, the equipments used and the user interface programmed are showed. The paper concludes by looking at the conclusions and future work in Section V.

2. Scenario and Protocols

The goal of the TetraMoD system is the creation of a media on demand delivery platform, based on multimedia broadcasting over DVB-T networks and the use of the TETRA network as interaction channel. The considered scenario (Fig. 1) uses the two networks altogether: DVB-T network, with media broadcasting capabilities, used for transmitting the feeds; and the TETRA network, used for requesting media presentations and transmission control. Both of these technologies are known for their ability to provide a high coverage as well as a high security degree for communications. Thus, this alternative is presented as an ideal solution for providing on demand downloading services

deployment as an added value facility for TETRA communications. The contents offered through this platform are heterogeneous, that is, they can be both data files as media streams. The latter could be both streaming of stored presentations as well as live feeds.

In our scenario, communication among TETRA devices is based on SDS messages since using the IP capabilities offered by the terminals prevent them from providing voice service.

The solution is designed in such manner that multiple users are able to receive the audiovisual information requested by one of them. This becomes particularly important for real-time video transmission.

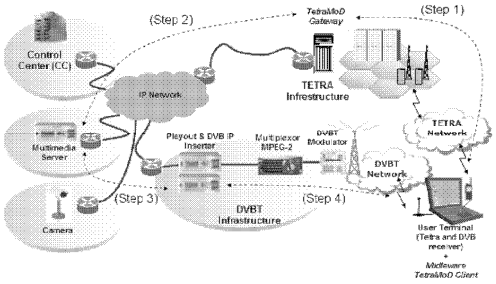


Figure 1: Multimedia System Architecture

A typical use case of this system is described next. Once an emergency call is received at the Control Centre, the research and operating units (fire-fighters, police, medical teams, etc) arrive and gain access to the disaster area. The rescue team has a terminal with access to TETRA and DVB-T networks, as well as the TetraMoD Client middleware installed on it. After inspecting the land and analysing the situation, it is crucial to possess the greater amount of information about the emergency area as possible in order to face the situation. Therefore, the rescue team will send a request to the Control Centre querying multimedia information about the emergency area-ranging from access routes, roads, buildings, maps, and so forth- using the TETRA network for this purpose so they can take advantage of its high coverage, dedicated bandwidth and secure access channel (Step 1). After that, the request is received and processed by the TetraMoD Gateway and redirected to the Content Server that actually provides the indicated content (Step 2). The needed multimedia information will be broadcasted over the DVB-T network and later received by the terminal client in order to improve making decisions (Steps 3 and 4).

The TetraMoD platform described is made up of several equipment elements, each one charged with a specific task in order to accomplish the steps needed to supply multimedia support for emergency teams. The IP Encapsulator (IPE from now on) has to gather the different media stream data coming from the Content Server and encapsulate them within elementary streams using the MPE protocol. The multiplexer combines a number of MPEG-2 elementary streams originated at different sources and arranges them within a single MPEG-2 Transport Stream. Last of all, the modulator generates a radiofrequency output signal for transmitting the desired Transport Stream in a given channel (Step 4).

On top of the platform architecture, the given solution uses IP protocols. In this manner, it provides a number of advantages like the following: a common layer to work with heterogeneous networks, interaction with currently available content delivering services based on IP as well as perspectives of imminent deployment over wireless networks. Also, the use of IP protocols is relevant regarding to resource management if we take into account that the available bandwidth for the service in the multiplex is limited, so use of the media on demand service by several users simultaneously can saturate the broadcast link. In this scenario, IP technology provides different mechanisms to optimize the bandwidth usage and control the QoS offered to the end user. The fact that neither DVB-T nor TETRA were originally conceived for carrying IP datagrams doesn't entail any inconvenience given that there have been defined standard mechanisms for data transmission over both of these technologies. A briefly description about the common protocols that have been used to develop this system follows.

- MPE (*Multi-Protocol Encapsulation*): A data link layer protocol defined by DVB which provides means to carry packet oriented protocols (like for instance IP) on top of MPEG-2 Transport Stream (TS). MPE uses MPEG-2 Private Table sections to carry the user datagrams.

- PEI (*Peripheral Equipment Interface*): A standard interface defined in TETRA for connecting a data terminal (e.g. a PC) to a TETRA radio terminal. It allows the data terminal equipment to use data transmission services over TETRA, like the transmission and reception of short data messages (SDS messages).

- SDP (*Session Description Protocol*) [6]: A standard protocol (RFC 2327) intended for describing multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. SDP does not provide the content of the media form itself but simply provides a negotiation between two end points to allow them to agree on a media type and format, for example during a RTSP session. It also reports connection information (IP addresses and ports), bandwidth information, etc. about the session being described.

3. Emergency Services

The system supports two kinds of services. On the one hand, it allows the delivery of video feeds in real-time (video streaming). On the other hand, a file transfer service is provided. An explanation of each one follows.

3.1. Video on Demand Service

With regard to the video streaming, the developed system allows the users of the client application to request both

stored video presentations as live feeds. These videos are transmitted using the RTP protocol over a DVB-T channel.

In an emergency situation, when a member of a Public Safety and Disaster Recovery (PSDR) organization asks for a surveillance video camera transmission, every other member of the same emergency response team will be able to see the images. In this way, when a disaster like a traffic collision took place, the emergency units going to the affected area could see the traffic cameras located near that zone so they could have a visual perspective in order to conduct their tactical and emergency operations in a more precise manner.

The major protocols used with this service are:

- RTSP (*Real Time Streaming Protocol*): An application protocol [7] for use in streaming media systems which allows a client to remotely control a streaming media server, issuing VCR-like commands such as "PLAY" and "PAUSE", and allowing time-based access to stored media presentations and live feeds. It's worth noting that the sending of streaming data itself is not part of the RTSP protocol, but data is sent out of band using for instance the standards-based RTP as the transport protocol for the actual audio/video streaming. Therefore, the RTSP protocol acts somewhat as a metadata channel.

- RTP/RTCP (*Real Time Transport Protocol/Real Time Control Protocol*): Standard protocols to enable multimedia communications over a best-effort network without a resource guarantee or QoS. RTP/RTCP provides sequence numbering and time stamping of multimedia packets as well as a feedback channel to obtain reception statistics. This pair of protocols has become, nowadays, the standard for multimedia streaming over IP networks.

3.2. File Transfer Service

The system also provides file sending capabilities for any kind of file format. This way, reception of both images as text documents amongst others is possible. In an emergency scenario, file downloading cannot be performed over the Internet due to coverage and connectivity limitations. Therefore, DVB technology is used instead and specifically FLUTE protocol is used for file downloading services. In our use case, for instance, the ambulance services can request a map of a secondary route where a traffic collision has taken place so they can rapidly access to the exact location.

Another clear example is information update performed by the police. An interesting application is the downloading of a robbed vehicles list on a daily basis so every day the police patrol can obtain an up-to-date registration numbers list using TETRA SDS requests. Since the file sending is multicast, the reception will be very fast and straightforward. The file can even be sent continuously so the downloading process doesn't require a previous request.

FLUTE (*File Delivery over Unidirectional Transport*) [8] is a protocol for the unidirectional delivery of files over the Internet, which is particularly suited to multicast networks. The protocol uses ALC (*Asynchronous Layered Coding*) [9], which is specifically designed to provide massive scalability using IP multicast. FLUTE was created to distribute files massively, offering reliability in the downloading process, and therefore it allows an asynchronous delivery of content to an unlimited number of receivers from one or various senders. ALC uses LCT (*Layered Coding Transport*) [10] to provide session management functionality, a congestion control block, as well as a FEC (*Forward Error Correction*) building block [11] to provide reliability.

A FLUTE session or file delivery session consists of a set of logically grouped ALC/LCT channels associated with a single source sending FLUTE packets. An ALC/LCT channel is defined by the combination of a sender and an address associated with the channel by the sender. A receiver joins to the channel to start receiving the data packets sent by the source, and leaves it when stops from receiving data packets or when wants to close the session.

Obviously, receivers must know the multicast addresses of the channels before they can join them and start receiving data packets. This and other relevant parameters of the multicast session are provided through out-of-band mechanisms, such as SDP, which has been mentioned previously.

One of the fundamental components FLUTE is based on is the File Delivery Table (FDT), which provides a means to describe various attributes associated with the files that are included within the file delivery session, for example the file location or its numeric identifier TOI. The FDT is delivered as FDT Instances, which are described in XML language, through FLUTE before the transmission of each file begins. Also, it is possible to send a general FDT at the beginning of the transmission which describes the properties of all session files.

In the shown scenario, the FDT allows checking in a quick manner whether a file requested by the client is found in the available files list maintained by the server.

When the server wants to transmit a FLUTE packet, it generates encoding symbols based on the object to be delivered using FEC codes and sends them in packets over the channels associated with the current session. The file segmentation process in encoding symbols is depicted in Figure 2.

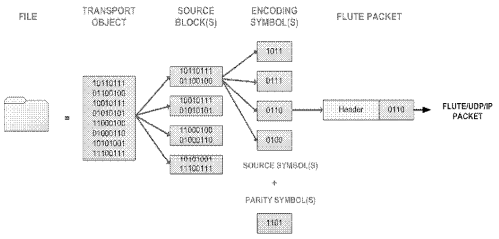


Figure 2: FLUTE packet building process

There are two kinds of sessions, File Delivery Sessions and File Carousels. The former consists on sending all the files associated with the session during a fixed time interval and then closing the session. In the latter, files are sent cyclically on a seamlessly ending loop. Thus, it provides a number of advantages, since the clients can connect to the session at any moment to receive any file. Furthermore, sessions can be static or dynamic (Fig. 3), depending on whether the contents of the session change during its lifetime.

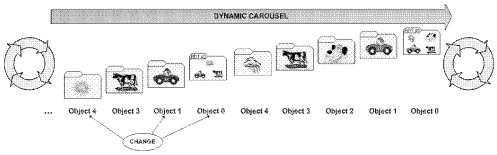


Figure 3: Dynamic carousel example

3.2.1. Grouping and Encoding

FLUTE protocol allows sending grouped files which have some common feature. Grouping files makes possible a clear and fast download for the client.

Grouping is performed by means of the File Delivery Table. The FDT holds every file property for the files that make up the carousel, such as their location, size, and so on. Also, it provides one or more fields in order to indicate the group each file belongs to.

In our scenario, wherever an emergency team is going to, it will be interested in gathering as much information about the target location as would be possible. Therefore, it can make a query related to that location that results in the server parsing and satisfying that request by sending all the files associated to the given group like maps, routes, and so forth. This way, a unique request will suffice in order to obtain all the needed information about the place.

On the other hand, one of the main features of FLUTE is the reliability that it provides for file downloading. This is achieved by means of two major mechanisms: on the one hand the continuous file forwarding using carousels and on the other hand the error control performed with FEC techniques.

Unlike video on demand, the file transmission has implicit encoding thanks to FLUTE, so employing encoding blocks foreign to the architecture is not needed. The encoding is carried out at the block level. As we can see in Figure 2, each source block is fragmented into source symbols. The encoding symbols –the source symbols plus the parity symbols related to each encoding– represent the payload of the FLUTE packet. By means of the FDT and FLUTE header of data packets, receivers know the encoding used as well as the block and encoding symbol length. At the receiving side, the decoder reconstructs each source block from a particular number of encoding symbols.

With the purpose of showing the advantages derived from use encoding, a number of tests have been carried out consisting on sending files with different redundancy through channels with different packet loss (using uniformly distributed errors). For these tests Reed-Solomon encoding has been used.

In Figure 4 we can see a comparative graphic that shows the needed number of transmissions for a given encoding type and redundancy employed, using different lossy channels with loss event rate ranging from 0,10% to 50%. A redundancy value of zero corresponds to no encoding used. As we can note, the graphic shows clearly the convenience of using some type of error correction, mainly when the files are sent over a channel with a high packet loss ratio. The benefits of utilizing encoding is such that with only a 5% of redundancy we can reduce by half the number of transmissions needed to receive a file successfully. As redundancy increases, the number of transmissions decreases exponentially until the ideal case is reached (only one transmission).

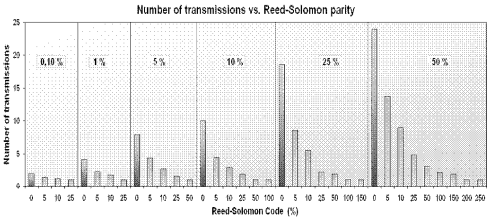


Figure 4: Number of cycles vs. coding

4. Implementation

An application targeting terminals like PCs with peripheral connectivity to TETRA and DVB-T networks has been developed with the purpose of validating the system proposed. TetraMoD Gateway provides a proxy service between the messages coming from clients with the TetraMoD middleware installed on and content servers. A description about TetraMoD Gateway and Client architectures, focused on implementation and functionality aspects, follows.

4.1. Middleware TetraMoD Client

The client middleware depicted in Fig.5 handles the media content queries as well as the DVB-T tuning, getting an IP flow encapsulated in a DVB-T channel as a result. These media queries are based on request/response messages. Since the communication channel to be used is the TETRA network, specifically the Short Data Service provided, these request/response pairs will be sent embedded within SDS messages.

The mentioned TetraMoD Client middleware has been programmed in Java. The reason for such a choice has been its ease of use for rapid prototyping of communication applications using TETRA devices via AT commands, thanks

to the Java Communications API. The facilities provided by TETRA terminals are accessed through the standard TETRA Peripheral Equipment Interface (PEI), which standardises the connection of the radio terminal to an external device, and supports data transmission between applications resident in the device and the connected TETRA radio terminal.

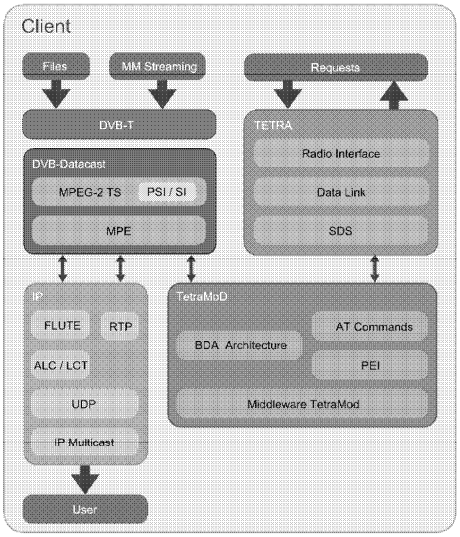


Figure 5: TetraMoD Client

On the other hand, the part of the system that performs the management, tuning and IP flow reception, has been developed using the TV and Broadcast Driver Architecture (BDA) API from Microsoft, thus programming in C++. The interaction between the BDA code and the Java middleware is done by means of a Java Native Interface (JNI) wrapper specially developed for this purpose.

The TetraMoD middleware permits requesting two kinds of services well differentiated: file transfer and audio/video streaming. File transfer is supported by FLUTE and ALC/LCT protocols, whereas media streaming is carried over the RTP protocol.

About this, the TetraMoD middleware incorporates the JVLC library from VideoLAN project, which allows media playback for many different formats and encodings.

4.2. TetraMoD Gateway

TetraMoD Gateway (Fig. 6) has been programmed in Java, due to its cross-platform nature and straightforward use that applies for rapid prototyping of client and gateway applications compliant with network protocols as RTSP. Furthermore, the Java Communications API extension has been used to support both TETRA mobile stations configuration and control as well of the middleware layer implementation. Like in the client, this is done by making use of the PEI interface found on TETRA devices for serial communication between the mobile station and the terminal equipment (e.g. a PC). Content requests are embedded in SDS messages whose format will be different according to the content itself (video or files) as it can be seen in the next section.

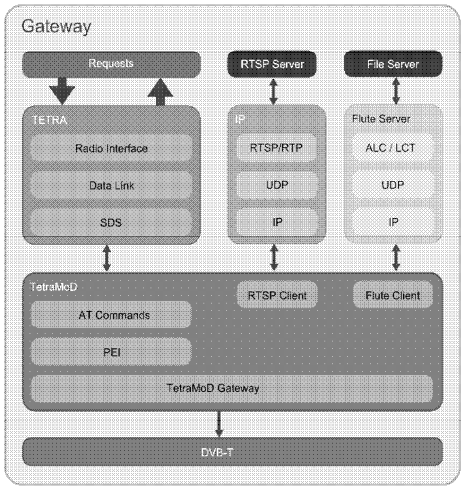


Figure 6: TetraMoD Gateway

In order to satisfy client requests the TetraMoD Gateway communicates with the Multimedia Server. As it is depicted in Fig. 6, the latter is in turn divided into two different servers, a RTSP Server for media streaming and a File Server for file transfer facilities. These servers will provide the media to be sent by the TetraMoD Gateway to the end user by means of DVB-T technology.

Also, the work of TetraMoD Gateway is to control the number of requests that are being satisfied at a given time and decide whether it is possible to accept new incoming client requests or not. In order to perform this task, TetraMoD Gateway parses the session description messages (using SDP protocol) sent from the Multimedia Server to find out the necessary bandwidth that must be allocated for each media stream and configure FLUTE server bit rate to accommodate state needs.

4.3. Description of end-to-end communication

1) Video on Demand Service

As it is shown in figure 7, communication between TetraMoD Gateway and the middleware layer is entirely performed through SDS messaging. These messages are processed by TetraMoD Gateway and then sent to the content server through normal RTSP over IP messages.

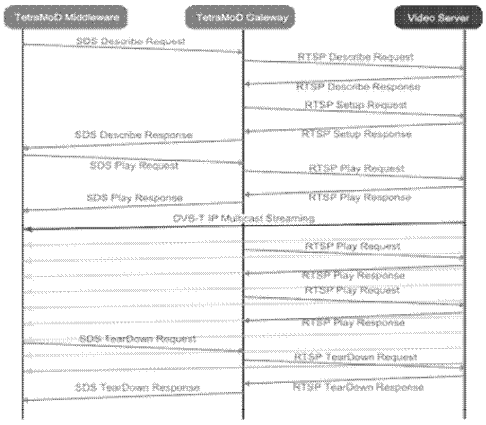


Figure 7: VoD end-to-end communication

Tetra SDS messages have a limited body size of 140 characters so they could not be able to fit within a single SDS message. To address this issue it has been defined a header for the SDS data body that let us handle RTSP messages longer than 140 characters. This header has two fields of one byte each one. The former is used to identify if the SDS has an entire RTSP message, if the RTSP message has been split into several SDS messages or if it is the last SDS of a split RTSP message. The second field has information about the fragment number when the RTSP message needs more than one SDS. SDS Messages are based on RTSP ones but we have introduced some changes in order to improve the communication (that we have dubbed RTSP Lite). Changes are related to unnecessary data elimination and data field abbreviation.

SDS Describe Request (using RTSP Lite)
D rtsp://www.dcom.upv.es/video.mov/
CS: 1
ACC: app/sdp

RTSP Describe Request
DESCRIBE rtsp://www.dcom.upv.es/video.mov/ RTSP/1.0
CSeq: 1
Accept: application/sdp

- -

RTSP Play Response
RTSP/1.0 200 OK
CSeq: 4
Date: Wed, 27 Feb 2008 14:17:23 GMT
Range: npt=0-328

SDS Play Response (using RTSP Lite)
200 4
R: npt=0-328

Figure 8: Examples of RTSP message and SDS message conversion

Also, TetraMoD system has added a number of modifications to RTSP in order to minimize the number of SDS messages sent by the TETRA devices. One of them involves doing the transport initialization, i.e. the RTSP SETUP phase, without the intervention of the client, as Figure 7 illustrates. For this purpose, the RTSP DESCRIBE Response that TetraMoD Gateway sends to the client has been modified so that it contains the transport mechanisms information previously agreed with the Multimedia Server. The major benefit of this behaviour is the increased efficiency in terms of TETRA transport usage and the ability of handling the addressing space to be used in DVB-T MPE elementary streams.

RTSP servers are expecting the clients for keeping alive the RTSP session by means of RTCP feedback messages. As this is not possible in the present system, it will be TetraMoD Gateway that keep alive the media presentation 'on air' using additional RTSP PLAY Requests sent to the content server at regular intervals during the RTSP session life.

The following table summarizes the traffic saving achieved both for bytes transmitted as for number of SDS messages sent when implementing the conversion system previously described. As a result we get a 44% compression rate regarding message size and a 52% saving in number of messages.

Table 1: Efficiency ratio of the message conversion

		SDS Body Size (bytes)		# SDS Messages	
		RTSP	RTSP Lite	RTSP	RTSP Lite
DESCRIBE	Request	85	57	1	1
DESCRIBE	Response	1047	637	8	5
SETUP	Request	108 x 2	-	1 x 2	-
SETUP	Response	151 x 2	-	2 x 2	-
PLAY	Request	93	72	1	1
PLAY	Response	81	21	1	1
TEARDOWN	Request	92	67	1	1
TEARDOWN	Response	24	7	1	1

	RTSP	RTSP Lite	%
Total Bytes	1940	861	44
Total SDS Messages	19	10	52

Nevertheless, the TETRA clients still will be able of closing the session by sending RTSP TEARDOWN Requests at any moment. Whenever the gateway receives live media streams requests for an ongoing RTSP session, it will do the negotiation process without querying the Video Server anymore. In this case, it will use the information obtained during the first media initialization phase to satisfy the subsequent client requests for the same media presentation. Moreover, the RTSP TEARDOWN Request sent by a given client will only be redirected to the Video Server if it's the last client interested in that media content, otherwise it will be discarded.

It's worth noting that the major bottleneck of the system is the available DVB-T channel bandwidth reserved for media streaming, so TetraMoD Gateway is perfectly capable and powerful enough for managing client connections without problems.

2) File Transfer Service

In the same way to the previous case, communication between user terminal and TetraMoD Gateway is performed using the Tetra SDS service, as it is shown in figure 9.

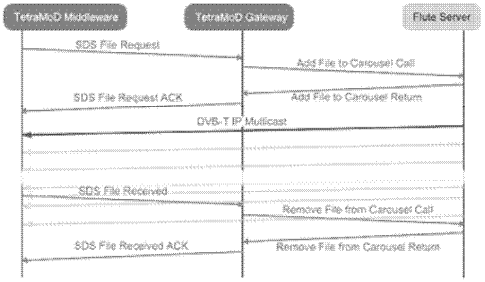


Figure 9: File Transfer end-to-end communication

The client sends a SDS File Request message querying the wanted file or file group. Then, the TetraMoD Gateway establishes a connection with the FLUTE Server requesting that the given file or file group be included within the file sending carousel. Hence, the FLUTE Server will transmit the carousel containing the file or group and all the clients who are tuned to that channel will be able of receiving the new content.

When the file has been successfully received, the client can send a message to the Gateway reporting this fact in order to the file could be removed from the carousel. The Gateway will check that the file is no longer expected by another client, and if this is the case, it will ask the File Server to modify the carousel.

As it is depicted in the figure, this model of Gateway-Server dialog doesn't need the Keep-alive messages to maintain the communication active.

4.4. User interface

In order to validate the system proposed it has been used the following equipment: a PC with a TETRA mobile station (EADS THR880i) attached to a serial port and a USB DVB-T Tuner (DIBcom DVB-T), an IP Encapsulator (SIDSA Vega IPDC), and a transmitter (ITIS V-XCAST DVB-T exciter). On the client side a PC application incorporates the middleware previously described.

The client application GUI (figures 10 and 11) has two tabs, each one corresponding to a different kind of content: video and file.

In the case of video streaming, the user can chose amongst the media available in the content list to the left of the application window. Once the media request has been satisfied, the video presentation will be shown by the media player on the right side (images, stored media presentations as well as live feeds).

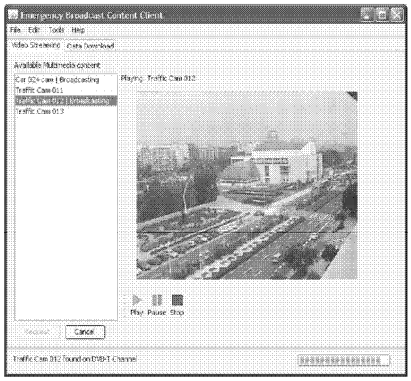


Figure10: User application (Video Streaming Service)

In the case of file transfer the process is analogous. The user just selects a single file or a complete group of them from the corresponding list in order to download them. This way, a thumbnail preview of the downloaded files will appear on the right pane of the application window.

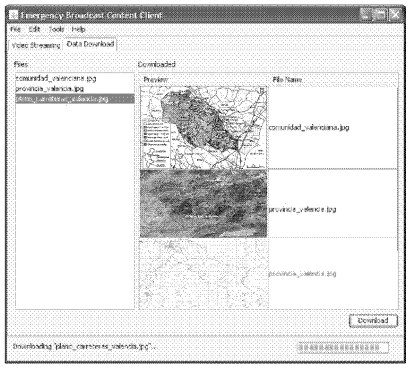


Figure 11: User application (File Transfer Service)

In both cases, upon selection, the corresponding SDS messages are generated and sent to the TetraMoD Gateway. Later on, once the client receives a number of SDS message containing the media description (IP address, ports, media type, encoding and format), it can request that the media streaming or file delivery starts. At the same time, the TetraMoD Client Middleware will tune the appropriate channel and extract the IP flow carried over.

If a video has been requested, the presentation playback will start and the image will be shown on screen, whereas if it is a file what has been requested a preview will be shown instead. The files will be saved in a directory previously defined by the application.

5. Conclusions

Today, emergency response teams need a secure, reliable and efficient communication system in order to carry out their tactical and emergency operations. TETRA networks are consolidated as a mature solution for Public Safety and Disaster Recovery agencies (like police and fire brigade) due to its reliability and confidentiality in the communication. Nevertheless, they are inefficient regarding to bandwidth provided, so a need for an added facility that satisfies this kind of services arises. The needed solution must provide high coverage, reliability and mainly high broadband capabilities. This way, it is possible to improve the service allowing both receptions of real-time video presentations as file downloading services in order to help to face the emergency situations.

Several solutions have been proposed in the last years. In this article we have opted to implement a solution based on the DVB-T standard, whose major features make it the more appropriate technology to provide broadband access to emergency response teams. The TetraMoD project has been successfully implemented emphasizing the efficiency of TETRA networks and the broadcast capabilities of DVB-T.

As future work, we present the need to enhance the security in the proposed platform communications. In this regard we propose the use of Raptor codes for communication reliability purposes. This type of encoding is recommended by DVB. Another line of future work is the development of client middleware for handheld devices, concretely for PDAs. These are a good option due to their continuous growth in memory, computational power and portability, being this last aspect very important in order to include the proposed system into body equipment.

Acknowledgments

This work was founded by Abertis Telecom -Development of Technology Group (DETEC) and Mobile Communications Services Group (SECOM).

References

[1] Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 2: Air Interface (AI), European Telecommunications Standards Institute Std. EN 300 392-2, Rev. 2.5.2, Nov. 2005, work item reference: REN/TETRA-03129.

[2] O. Andrisano, A. Bazzi, A. Giorgetti, G. Pasolini, and V. Schena, "*Audio-Video Services for Emergency Scenarios over Heterogeneous Satellite-WiFi Networks*", in Proceedings of the IEEE Wireless Rural and Emergency Communications Conference (WRECON), 2007, Rome (Italy).

[3] IBBT. (2007, Jan.) Advanced disaster architecture with mobility optimizations. [Online]. Available: https://projects.ibbt.be/adamo/

[4] M. Berioli, N. Courville and M. Werner, "*Integrating Satellite and Terrestrial Technologies for Emergency Communications: the WISECOM Project*", in Proceedings of the Fourth International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QSHINE), 2007, Vancouver (Canada).

[5] ETSI EN 300 744 – "DVB-T, Framing structure, channel coding and modulation for digital terrestrial television".

[6] M. Handley and V. Jacobson, "SDP: Session Description Protocol". RFC 2327 (1998)

[7] H. Schulzrinne, A. Rao and R. Lanphier, RFC 2326 "Real Time Streaming Protocol (RTSP)". RFC 2326 (1998)

[8] T. Paila, M. Luby, R. Lehtonen, V. Roca V and R. Walsh, "File Delivery over Unidirectional Transport (FLUTE)". RFC 3926 (2004)

[9] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo and J. Crowcroft, "Asynchronous Layered Coding (ALC) Protocol Instantiation". RFC 3450 (2002)

[10] M. Luby, J. Gemmell, L. Vicisano, L. Rizzo, M. Handley and J. Crowcroft, "Layered Coding Transport (LCT) Building Block". RFC 3451 (2002)

[11] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley and J. Crowcroft, "Forward Error Correction (FEC) Building Block". RFC 3452 (2002)

Modela-TV: Service Personalization and Business Model Management for Mobile TV

Francisco Fraile
Multimedia Communications Group
iTEAM, Polytechnic University of Valencia (UPV), Spain
Phone: 34-963879717
ffraile@iteam.upv.es

Ismael de Fez
Multimedia Communications Group
iTEAM, Polytechnic University of Valencia (UPV), Spain
Phone: 34-963879717
isdefez@iteam.upv.es

Juan Carlos Guerri
Multimedia Communications Group
iTEAM, Polytechnic University of Valencia (UPV), Spain
Phone: 34-963879717
jcguerri@dcom.upv.es

ABSTRACT

This paper presents Modela-TV, a research project which investigates how free-to-air file download services can be the basis of new business models in mobile multimedia content delivery networks. Hereby, we present the proposed system architecture, together with implementation guidelines for the main technologies involved. A novel method for the personalization of broadcast file download services, based on cache management, is introduced. This represents one of the main outcomes of the project, as personalization is considered a key feature in mobile TV services.

Categories and Subject Descriptors

D.2.6 [**Software Engineering**]: Programming Environments - *Interactive environments*. E.5 [**Files**]: *Optimization, organization/structure*. H.3.3 [**Information Search and Retrieval**]: *Information filtering*.

General Terms

Design, Performance, Management.

Keywords

Mobile TV, Service Prototype, Personalization, Filecasting, Content management, Service Guide.

1. INTRODUCTION

Current Mobile TV revenue streams are primarily based on billing customers, either for content and transmission, as in the case of On Demand Streaming Services deployed by 3G Mobile Network Operators, or for access to a bouquet of premium TV channels, as in the case of Mobile Broadcast Services. In the traditional (non-mobile) television business, main revenue streams are generated by advertisement on free-to-air services, but also by premium and pay-per-view based business models. Enabling such a scenario in the Mobile TV landscape is regarded as a key factor to boost the penetration of mobile TV services. Current revenue streams are too weak to meet the expectations of all the agents in the value chain. A prove of this fact is that most of the current incomes of mobile streaming services are held by content providers.

On the other hand, the availability of content for download on the Internet has shifted user habits and expectations on media services. Regardless of legal implications, users regard content as easily available and affordable (if not free). In a mobile scenario, this implies that users expect receiving content of their personal interest during the short time they are connected to the service.

For these reasons, the project "Service Personalization and Business Model Management for Mobile TV (Modela-TV)" proposes a Mobile TV service which allows service operators to use different business models (free-to-air, pay, premium…) simultaneously when offering content to mobile users. The service takes into account user expectations with regards to content availability and cost. The aim of the project is to develop a business model management platform for mobile multimedia delivery services through which, a) service operators can enable several revenue streams simultaneously and b) users can access personalized content at an affordable cost.

2. TECHNOLOGIES

The Modela-TV platform manages several technologies but relies mainly on two: on one hand, FLUTE, for the reliable distribution of files over the mobile broadcast/multicast network. On the other hand, the platform also manages a DRM subsystem in order to protect the intellectual property of the content to be delivered. These systems provide an adequate technical framework to deal with the commercialization of protected content. New ways of delivering content are introduced, which are meant to adapt the usage of the broadcast network in order to match the expectations of the users.

2.1 FLUTE

FLUTE (*File Delivery over Unidirectional Transport*) [1] is a protocol for the unidirectional delivery of files over the Internet, which is particularly suited to multicast networks. The protocol uses ALC (*Asynchronous Layered Coding*) for a reliable and scalable transport of the multicast files. Furthermore, ALC implements a Forward Error Correction block to accomplish reliable multicast.

A FLUTE session consists of a set of logically grouped ALC channels associated with a single source sending FLUTE packets. Each FLUTE session is uniquely identified by the source IP address and the Transport Session Identifier (TSI). A receiver joins the channel to start receiving the data packets sent by the source. Previously, receivers must know the multicast addresses of the channels using out-of-band mechanisms such as SDP *(Session Description Protocol)*.

There are two kinds of FLUTE sessions, File Delivery Sessions and File Carousels. The latter consists of a cyclical transmission of all files to be transmitted on a seamlessly endless loop and it is the one normally used, since cyclical transmission provides a

reliable mechanism to receive files correctly. Furthermore, sessions can be static or dynamic, depending on whether the contents of the session change during its lifetime.

One of the fundamental components of FLUTE is the File Delivery Table (FDT), which provides a means to describe various attributes associated with the files that are included within the session, for example the file *Content-Location* (URL or URI of the item) or its numeric identifier TOI (Transport Object Identifier), used by the ALC client to filter the packets corresponding to the wanted files. The FDT is delivered as FDT Instances, which are XML fragments providing descriptions of one or several files over ALC with TOI = 0. Receivers need parsing the FDT Instance that describes a file before it can start the download.

DVB IPDC (IP Datacast) [2] defines file grouping in FLUTE as a mechanism to download together files that have some kind of dependency at application level. This grouping is performed by means of the elements *Group* contained in the *file* element of the FDT Instance that describes a file. For each *file* element, one or several elements of type *Group* may be defined. The FLUTE clients that start downloading a file with a given *Group* element must start download all files with the same *Group* tag. Thus, file grouping enables an efficient caching of files.

2.2 DRM

The term DRM (*Digital Rights Management*) refers to a series of technologies that enable the consumption of licensed digital content in a controlled environment. The OMA BCAST Service and Content Protection [3] DRM Profile provides such a technological framework in order to support the delivery of licensed content over broadcast networks under a variety of business models. According to these specifications, the usage of DRM in Modela-TV is explained below.

Modela-TV adopts the *Subscriber Group Addressing* feature, which allows to define the exact group of broadcast receivers that will be able to access the protected content. Access to content is granted according to the terms and conditions defined by the content owner for the specific subscriber group (e.g. content preview, advertisement placement). *Subscriber Group Addressing* consists of encrypting ROs (*Right Objects*) with UGKs (*Unique Group Keys*). ROs are messages that express these access terms and provide the means to access the content. Thus, in the registration phase, all authorized receivers in a *Subscriber Group* are provided with the same UGK which will allow them to decrypt a series of ROs.

Furthermore, in Modela-TV the content is encrypted with CEKs (*Content Encryption Keys*) and encapsulated in *dcf* (*DRM Content Format*) files. Upon reception of a *dcf* file, a receiver will use its keys (i.e. a set of UGK) to decrypt the ROs within the *dcf* file. If a RO is decrypted, the receiver will use the CEK to decrypt the content and display it to the user, according to the rules expressed in the RO.

2.3 ESG

The Electronic Service Guide (ESG) is a file download service used to provide users and terminals with a description of the services available on an IP platform (similar to Electronic Program Guides of Digital TV Services).

Modela-TV adopts the usage of OMA BCAST ESG standard [4] as follows: the *Access* fragments of the ESG are used to provide the FLUTE session information for File Download services (referencing a FLUTE SDP fragment). The *Schedule* fragments of a content file instantiate the element *ContentLocation,* which value is equal to that of the attribute *Content-Location* of the FDT Instance of the file, thus completing the access information required by the FLUTE client to fetch the file from the carousel.

Moreover, the *Access* fragments contain a *KeyManagementSystem* element which indicates that the DRM profile is used. The *RightsIssuerURI* attribute provides receivers with information to obtain rights associated to the content item. Additionally, *PurchaseChannel* fragments establish a relationship between the URIs and URLs of RIs, so that receivers can contact RIs if the user wants to acquire new rights.

Lastly, each file of the carousel is associated with a *Content* fragment which provides descriptive information such as parental rating, textual descriptions or TV-Anytime [5] genre descriptions. This descriptive information is used to implement personalization features in the service, as explained later.

3. ARCHITECTURE

The figure 1 shows the general architecture of the system, showing the most relevant technologies involved in the project.

The *Business Model Manager* is the main entity of the architecture and the interface between the platform and the operator. It manages the content ingestion and the Digital Rights associated to each media item. Through the *Business Model Manager*, the operator establishes the relationships of the content items with each Subscriber Group. This way, the management of the subscriber data base, the repository of content and the associated metadata (user profiling, purchase information and content description) are also in control of the operator through the user interface of the *Business Model Manager*. According to the setup provided by the operator, the *DRM subsystem* encrypts the content items and produces the *DRM Content Format* (dcf) files which are then aggregated to the free-to-air broadcast by the *Filecast Server*. The latter also delivers the metadata (content descriptions, purchase information, etc.) to users in the service area by aggregating the ESG of the service.

The *Filecast Server* is the entity that deals with the delivery of the content through the mobile broadcast network. The *Filecast Server* queries the *Business Model Manager* to obtain the metadata associated to each content item and aggregates this information in the generated ESG. The ESG is broadcasted and then received by the client application to discover which content items are available and their characteristics. Furthermore, the *Filecast Server* also manages the distribution of the content associated to each service by managing the configuration of the FLUTE carousel sessions and the definition of the associated File Delivery Tables. The software supports the definition of File Groups which, as described in the FLUTE section, can be used to accelerate the acquisition of a compilation of content items.

In the current version, the *Filecast Server* provides a constant bit rate IP flow transporting the FLUTE sessions and the File Carousel Manager is responsible for scheduling the transmission of files. An optimal transmission schedule minimizes the overall access time to files in the service area [6]. In order to accomplish

this, instances of File Carousel Manager can configure the schedule of the transmission on a per FLUTE packet basis, thus enabling the implementation of advanced scheduling policies.

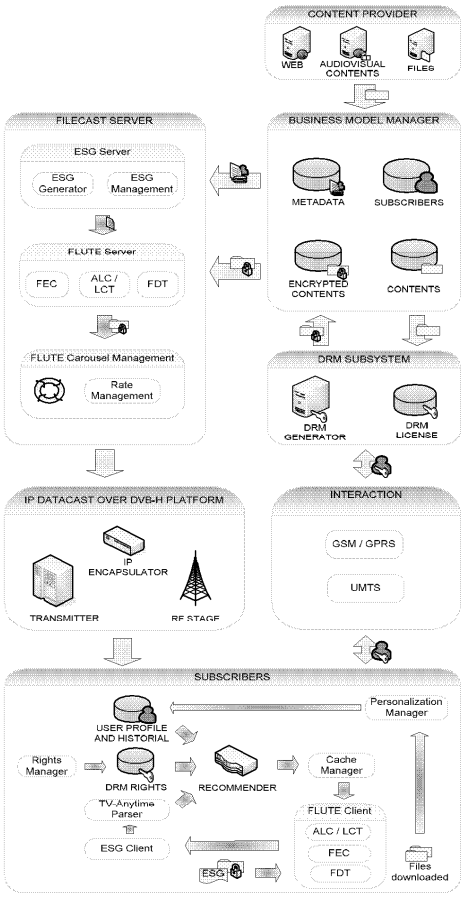


Figure 1. Modela-TV general architecture

In the case of study, the content is delivered by means of a *DVB-H distribution platform*. Moreover, the *Filecast Server* aggregates the FLUTE sessions which correspond to the content delivery services and their Service Guide as a set of IP multicast streams that are encapsulated into an MPEG Transport Stream. The latter is broadcasted by the DVB-H air interface. The IP Encapsulator sets the configuration of the MPE encapsulation (time slicing and MPE-FEC parameters) and aggregates the metadata tables (PSI/SI) which are necessary to extract the IP data from the MPEG-TS by the receivers.

The client application provides *subscribers* with a user interface which discovers and lists all the content available through the service. Once the DVB-H receiver tunes to the frequency that transports the services, the client application receives the Service Guide and presents a list of the available content for download. When the user selects a content item, this is downloaded to the terminal as a background service. The client application also allows entering user preferences and downloading content accordingly. In order to do this, the client application interprets the content descriptions provided through the ESG and selects the media items which better suite the user preferences.

If *interaction* is required, an interactive channel like GPRS or UMTS can be used. As explained, the ESG provides a Right Issuer URL to enable the communication with the system for the acquisition of digital rights, in order to gain access to the desired content. The request and later acquisition of these rights is carried out by this block, since DVB-H does not provide a return channel.

4. PERSONALIZATION

Personalization is one of the most important design aspects of mobile multimedia services [7] and therefore, it represents one of the main goals of the project. Personalization provides many advantages to the user, for instance the automatic discovery of interesting content. The perception of the service adapting to the user preferences and needs encourages a positive experience, most significantly when accessed from personal devices such as mobile phones.

Personalization is also a great asset for content distributors. An efficient and functional application favors the consumption of content, thus increasing the market share of mobile content production. Since users receive content according to their preferences, content distributors can rely on the platform delivering the content only to their target audience and advertisers can improve the impact of their campaigns.

In Modela-TV, personalization is achieved by means of a recommender on top of the DVB-H middleware that interacts with the ESG client and the cache manager as explained hereafter.

First, terminals only store in cache those files for which the user has associated access rights, thanks to the coupling between the DRM subsystem and the *Filecast Server*. This is accomplished by relating the Subscribers Groups to the values of the *Group* elements that are included in the FDT. As explained, the UGKs are used to encrypt Right Objects that target a group of users, whereas all files with common *Group* tags are fetched by the FLUTE client after a single download instruction. So, if a user decides to download a file, the application will first verify that it has access to it. If negative, it will prompt the user to acquire access rights (using the *Purchase Channel* information included in the Service Guide). If positive, the FLUTE client will start downloading that file and all the files with the same *Group* tag, which will be stored in cache.

Accordingly, the average access time to any other file for which the user has associated rights is reduced, since all content items associated to the user's categories of subscriber are cached automatically by the FLUTE client (recall that all files in a Group are downloaded by a single download operation on any of them). If the size of the set of files to cache is bigger than the size of the

allocated memory, then the caching replacement policy will determine which files remain in cache. In Modela-TV the caching replacement policies are based on a method named *PIX* (*Probability Inverse Transmission Rate*) and described below.

In a broadcast scenario, the cache management algorithm [6] must store in cache the objects with higher ratio between the future probability of access, P_i, and the object transmission rate, since the value of storing an object in cache is inversely proportional to its rate. This is due to the fact that, if an object has a long transmission cycle, then the access time of that file is drastically reduced in the event of a cache hit. Thus, for each incoming file, the cache replacement policy computes its *PIX* ratio, compares it to the files stored in cache and discards the file with lower *PIX*.

In order to implement *PIX* based cache replacement policies, an estimation of the local probability of access is needed. In our case, this is provided by the recommendation tool. Recommenders are software tools that estimate the usefulness of content items in a catalogue for a user [8]. In this proposal, it is assumed that the usefulness of a content item is directly proportional to its future probability of access P_i. Upon this assumption, recommendation tools can be used to derive an estimation of the future probability of access to items. This way, for each content item in the carousel, the recommender compares the TV-Anytime descriptions of the content with a user profile that represents the preferences of the user, in order to derive an estimation of P_i. With this information, the cache manager stores in cache memory the files with a higher *PIX* ratio.

The *Personalization Manager* tracks the content consumed by the user, creates user profiles and updates history data so that the recommender can estimate P_i accurately from realistic information about the consumer preferences.

5. CONCLUSIONS AND FUTURE WORK

Modela TV provides with a platform for the deployment of personalized multimedia content delivery services in mobile multicast networks. The design of the system architecture of Modela-TV has taken into account the expectations of users with regards to personalization, download speed and content availability, without compromising the interests of content license holders. Furthermore, service operators are provided with means to monetize the content through a variety of revenue streams at a very affordable cost, compared to current mobile multimedia content delivery services.

Modela-TV also proposes a new method for the personalization of multicast file download services, based on the use of FLUTE groups and content descriptions to optimize the usage of the cache memory allocated for the service.

A prototype of the platform has been implemented and it has been tested in a test laboratory assembled at the Institute of Telecommunications and Multimedia Applications (iTEAM) of the Polytechnic University of Valencia, as illustrates the figure 2.

In this figure we can see the equipment employed to perform the architecture tests and the running application. The current hardware for the client prototype consists of PDA with a SDIO DVB-H receiver (SIDSA). Application GUI is shown on the screen. Once the client is successfully connected to an IP multicast address, port and TSI, the user can start downloading files and groups in the carousel.

Beside the laboratory tests, the service is going to be tested by means of a pilot transmitter installed in the University Campus. The Radio Frequency Planning of the test transmitter has been performed with iTEAM´s planning tool, guaranteeing coverage in all the University Campus. The pilot will host field trials and end user trials.

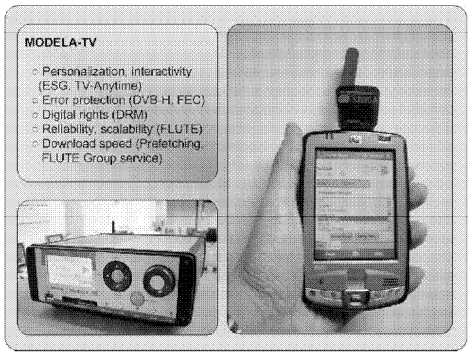


Figure 2. Modela-TV Application

Current research is focused on the study of optimal file schedule configurations that minimize the access time to files. Also, the project will evaluate how the design of the recommender affects the performance of the cache manager.

6. ACKNOWLEDGMENTS

This project has been financed since 2007 by the Spanish Ministry of Industry, Tourism and Commerce, within the program "Avanza I+D", as part of the financing of the National Plan of Scientific Research, Development and Technologic Innovation. The project has also been supported by Interactive TV Arena, in the framework of the co-operation agreement held with the iTEAM.

7. REFERENCES

[1] Paila, T., Luby, M., Lehtonen, R., Roca V., Walsh, R.: File Delivery over Unidirectional Transport. RFC 3926 (2004).

[2] ETSI, TS 102 472 v1.2.1: Digital Video Broadcasting (DVB); IP Datacast over DVB-H, Content Delivery Protocols" (2006)

[3] Open Mobile Alliance: Service and Content Protection for Mobile Broadcast Services V.1.1 (2008).

[4] Open Mobile Alliance: Service Guide for Mobile Broadcast Services, Candidate Version 1.0 (2008).

[5] ETSI, TS 102 822-3-1 v1.4.1: Broadcast and On-line Services: Search, select and rightful use of content on personal storage systems ("TV-Anytime") (2007).

[6] S. Acharya, R. Alonso, M. Franklin and S. Zdonik: Broadcast disks: data management for asymmetric communication environments, ACM SIGMOD Conf., 1995, pp. 199-210.

[7] K. Chorianopoulos. Personalized and mobile digital TV applications. Multimedia Tools and Applications, Springer, 2007.

[8] G. Adomavicius, A. Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-Of-The-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 6 (June 2005), pp. 734-759.

1277

Personalized Multimedia Touristic Services for hybrid broadcast/broadband mobile receivers

Alberto Gil*, Francisco Fraile§, Manuel Ramos*, Ismael de Fez§, Juan Carlos Guerri§

* University of Vigo, Spain § Polytechnic University of Valencia, Spain

***Abstract*—This paper presents the topology of a multimedia tourism service that targets mobile devices with broadband and broadcast access capabilities. The solution adopted relies on existing mobile TV standards, but introduces features that enable the provision of generic multimedia services on top of hybrid mobile networks**

I. INTRODUCTION

The introduction of mobile TV services is following a much slower pace than initially predicted. However, a variety of mobile broadcast networks and technologies are present all over the world. Among these, Hybrid Broadband Broadcast architectures, such as OMA BCAST [1], use mobile broadcast access in combination with mobile broadband access to enable the provision of rich interactive mobile TV services. However, the difficulties inherent to mobile reception make dedicated mobile broadcast networks expensive. Thus, there is a need for new services that benefit from mobile broadcast network resources while generating new revenue streams to ease the monetization of network Capital Expenditures and in turn to encourage the mobile TV market.

On the other hand, services for mobile phones are moving towards a new landscape where mobile applications are seen as highly specialized pieces of software that are independent of each other. In this new scenario, it is interesting to regard new mobile broadcast services not as interactive applications necessarily coupled to the mobile TV offer, but as independent services that bring added value to users while creating new revenue streams in the mobile broadcasting value chain.

This paper describes the topology of a multimedia tourism information service targeting mobile devices that implement both broadcast (DVB-H) and broadband (HSDPA) access. We describe how to use standardized interfaces originally defined for interactive mobile TV broadcasting to build personalized mobile multimedia services that do not deal with TV contents or are not necessarily coupled to a mobile TV service bundle.

II. SERVICE DESCRIPTION

The purpose of the service under study is to provide personalized touristic information to visitors during their stay in a city resort. The service operator is a local tourist agency that, by means of this service, is able to aggregate multimedia content with tourism information into the hybrid broadcast/broadband distribution platform.

Apart from content items, the operator can also include services of the mobile telephony network, such as sending message templates, get contact information, access external links, make automated reservations, buy tickets or hire a baby-sitter via web services.

Tourists access the service through terminals capable of accessing both networks. Upon arrival to the destination resort, the user downloads and installs an ad-hoc client application, meant to access and render the items managed by the platform. This application then connects to a local server and downloads a tourist guide through which users can browse item descriptions.

The multimedia content items within the tourist guide are not delivered together with the tourist guide. Instead, the platform uses the broadcast access to distribute the most popular items. The rest of the content items are available through the broadband access, together with the interactive services.

To accomplish the content distribution over both networks, the tourist guide, which complies with the OMA BCAST *InteractivityMediaDocuments* (IMD) XML document format, provides an access URI for every content item in the guide. In our implementation, terminals are able to resolve this URI into either access information to a GZIP encoded file in a broadcast FLUTE session or to a GZIP file in an HTTP server. Each GZIP file contains XHTML presentation markup together with the multimedia content (e.g. pictures or videos) embedded into that particular tourist multimedia item.

III. PERSONALIZATION

The success of the hybrid content distribution scheme promoted in this project is largely dependant on its effectiveness to satisfy users. This is especially difficult in major destinations due to the vast and heterogeneous touristic offering, parallel to the visitors' demographic variety. This way, it is not easy to provide a complete and general offering of touristic resources without overwhelming the users with a countless collection that makes it difficult to take advantage of the service.

Consequently, a recommendation module in charge of giving priority to resources depending on the visitors' characteristics is a must in this kind of framework where the user terminal is individualized (though it may gather information about the visitors' travelling companion).

To this extent, this project implements a selection mechanism that performs an ordering of touristic resources according to their interest for a given visitor. The recommendation engine is an adaptation of the well-known recommender AVATAR [2][2], extensively tested in different environments for TV content recommendations or for personalized composition of distance learning courses over TV.

In this tourism context, the recommendation engine has been adapted to estimate the adequacy of a resource for a visitor, according to the available information:

- The characteristics of the touristic resource, taking into account the seasonal constraints.
- Demographic data about the visitors and their preferences.
- Their explicit ratings for previously visited resources. This includes resources located in other destinations as long as they share the recommendation platform.
- Implicit conclusions about their preferences, extracted from their historical behaviour regarding services contracted through the platform (tickets, restaurants, reservations…).

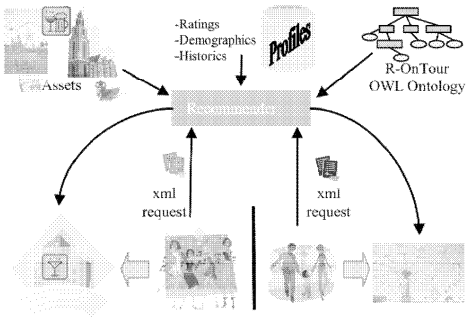


Fig 1. Recommendation system

The recommendation service implements a hybrid strategy composed of two stages: content-based (comparing resources to users) and collaborative filtering (comparing users among them). The kernel algorithm is based on the AVATAR original strategy, fully described in [2]. Starting from the aforementioned techniques that compose the hybrid approach, the suggestions are enhanced with the additional discovery of new significant relationships by means of techniques extracted from the semantic web field. To this extent, all the information characterizing the touristic resources has been organized by means of an OWL ontology named R-OnTour, an ad-hoc extension of the well-known OnTour ontology developed in the OnTour project [3].

By means of the Java API Jena, the recommendation engine explores and weights relationships derived from the properties that characterize the resources in the ontology, searching for semantic links that relate the local tourist resources to those ones positively rated by the user in the past or inferred to be appealing to the user by the system.

IV. LATENCY

In order to provide a satisfactory experience, it is also important to handle the latency of the file delivery over the hybrid distribution network.

Looking at the broadband access, the average latency is mainly dependent of the network delay in overload conditions. To prevent the occurrence of overload events, a back off timing mechanism that distributes the user requests over time is introduced. Most popular files will need larger back off timing windows to level out the access peaks, thus experiencing longer access times.

Hence, the placement of popular files in the broadcast access helps keep the service latency under control, especially in tourist areas where a high density of users per cell is expected. As explained in [4], the average access time to files in the broadcast service area can be optimized, given that an empiric measure of the popularity of files is available. In our implementation, the recommender engine establishes which files compose the broadcast carousel and their respective cycle time. This way, files with higher ratings have a lower cycle time. Similarly, the back off timing parameters of the files in the broadband access are also given by their respective ratings, as an estimate of their popularity, to minimize the probability of network overload.

On reception of the IMD file through the interaction channel, clients start to prefetch the GZIP files from both networks and to store them in cache as the user interacts with the personalized tourist guide. Then, it is necessary to account for the local probability of access of each user in the cache management policy. Again, the recommender software is used to assess the value of items in cache memory. This value is proportional to the local probability of access and to the time cost associated to the file download. The cache manager uses the recommender ratings to estimate the local probability of access whereas the cost for accessing each file is assessed from the available metadata.

V. CONCLUSIONS

This paper has presented a novel tourist information service provisioned through state-of-the-art hybrid networks. Terminals use both broadcast and broadband network resources to access the tourist multimedia information assets. To improve usability and avoid overwhelming the user with information, the platform provides users with personalized tourist guides according to their preferences. The recommender module information is in turn used to minimize the hybrid network latency experienced by users.

ACKNOWLEDGMENTS

This work has been funded by the Spanish Ministry of Industry under project Redes Híbridas. The authors would like to thank the rest of the members of the project consortium for their comments and suggestions.

REFERENCES

[1] Open Mobile Alliance™, OMA-AD- BCAST-V1_0 "Mobile Broadcast Services Architecture", 12 February 2009.

[2] Y. Blanco, J. Pazos, M. López, A. Gil, M. Ramos, "AVATAR: an improved solution for personalized TV based on semantic inference", *IEEE Transactions on Consumer Electronics*, vol. 52, pp. 223-231,2006.

[3] Katharina Siorpaes and Daniel Bachlechner, OnTour: Tourism Information Retrieval based on YARS, 3rd European Semantic Web Conference (ESWC-2006), Budva (Montenegro), 11 - 14 June, 2006.

[4] G. Zhiqi, Y. Songyu, Z. Wengjun "Using object multiplex technique in data broadcast on digital CATV channel" *IEEE Transaction on Broadcasting*, Vol. 50, Issue 2, June 2004.

Personalized Multimedia Touristic Services for hybrid broadcast/broadband mobile receivers

Alberto Gil, Francisco Fraile, *Member*, IEEE, Manuel Ramos, Ismael de Fez, Juan Carlos Guerri

Abstract — *This paper presents the topology of a multimedia tourism service that targets mobile devices equipped with broadband and either multicast or broadcast access capabilities. Multicast / Broadcast access is used to handle the delivery of multimedia data to many simultaneous users in the service area. The solution adopted is based on existing standards for mobile TV services, but introducing features that enable the provision of generic multimedia services on top of mobile hybrid broadband and broadcast networks. Furthermore, this paper shows how to use standardized interfaces originally defined for interactive mobile TV broadcasting to build personalized mobile multimedia services that do not deal with TV contents or are not necessarily coupled to a mobile TV service bundle.* [1]

Index Terms — Multimedia Tourism Services, Mobile TV, Hybrid Networks, Personalization

I. INTRODUCTION

Mobile streaming media applications set very high standards on mobile data networks transmission rates. With point-to-point streaming solutions, the pressure on the network is felt more intensely in certain hotspot locations with a high concentration of users. When this occurs, the mobile network collapses due to congestion, unless there is sufficient network infrastructure to cope with all the media requests. A solution is to use multicast or broadcast access in these areas where the density of streaming media users exceeds the number of simultaneous mobile point-to-point connections that can be served at a reasonable cost.

This is one of the main reasons behind the appearance of a series of mobile broadcast technologies, such as DVB-H, MediaFLO™ or T-DMB, meant for the provisioning of mobile TV services in wide areas. Another example is Multimedia Broadcast Multicast Service (MBMS), an amendment to provide broadcast and multicast services on top of GSM and UMTS networks.

Although these technologies are present in a variety of markets, mobile broadcast and multicast are rather expensive and there is a need for new services that benefit from mobile broadcast network resources while generating new revenue streams, in order to ease the monetization of network Capital Expenditures and in turn to encourage the mobile TV market, reckoning that mobile TV is the drive of this technologic development.

Touristic mobile multimedia services are good candidates for enhancing the mobile broadcast / multicast content offer, as certain tourism information items are relevant to most visitors of a touristic destination. Furthermore, in most cases, there is a lot of touristic multimedia content that tourism agencies would like to offer to their visitors.

However, a linear mobile TV channel may not be the best solution for a tourism multimedia service. Firstly, the content that is to be offered may not be video, as there are other kinds of content assets, like pictures or text, which could be used by the service. Second, service operators may wish to associate to certain content items some communication services, such as messaging (SMS or e-mail) or voice services, in order to enable automatic reservations, as an example. Also, it may not be convenient to arrange the content delivery in a linear program grid, as users might expect some degree of service personalization and receive not just general interest information but also information tailored to their user profile.

All these features can be implemented by associating interactive services to a mobile TV channel offer, as proposed by the OMA BCAST Services standard specification. Through the Electronic Service Guide (ESG), an operator can announce the offered services and the way to access them. The associated services can provide interactive elements and downloadable content items to bring in the missing interactivity and personalization features.

On the other hand, services for mobile phones are moving towards a new landscape where mobile applications are seen as highly specialized pieces of software that are independent of each other. In this new scenario, it is interesting to regard new mobile broadcast services not as interactive applications necessarily coupled to the mobile TV offer, but as independent services that bring added value to users while creating new revenue streams in the mobile broadcasting value chain.

This paper describes a tourism service for mobile devices that:

- Presents personalized tourism guides to visitors. The guides include multimedia content and communication services. The service can be deployed in large areas (a city, a region, a country) or in hotspots (a tourist resort).
- Implements a personalization engine that adapts the content in the guide to the profile of the visitor. The personalization

[1] This work was supported in part by the Spanish Ministry of Industry under project TSI-020302-2008-94.

A. Gil and M. Ramos are with the Department of Telematic Engineering of the University of Vigo, Vigo, Spain (e-mail: agil@det.uvigo.es)

F. Fraile is with the Institute of Telecommunications and Multimedia Applications of the Polytechnic University of Valencia, Valencia, Spain and with Interactive TV Arena, Gävle, Sweden (e-mail: ffraile@iteam.upv.es)

J. C. Guerri and I. de Fez are with the Institute of Telecommunications and Multimedia Applications of the Polytechnic University of Valencia, Valencia, Spain (e-mail: jcguerri@dcom.upv.es)

is a critical aspect in a tourism guide service, since the content offer may be too large for browsing in a mobile device and affect negatively the user experience.

- Uses both broadband and broadcast networks to deliver content to mobile terminals. The content delivery is based on IP protocols, and it is compatible with both mobile broadcast and mobile multicast technologies. The application middleware only requires IP multicast support.
- Is compatible with mobile broadcast standard specifications and therefore, it is useful to monetize the investments in mobile broadcast network infrastructure. Moreover, the service is available to OMA BCAST mobile terminals.
- Can be associated to a mobile TV service offer, as an interactive TV service. Also, it is possible to install a standalone client application in terminals that do not support interactive mobile TV services..

The next section presents a general description of the service. Section III describes the platform architecture, together with the main technologies involved. Section IV goes in depth in implementation details. Section V presents some related work. And finally, Section VI draws some conclusions.

II. Service Description

The purpose of this project is to provide personalized touristic information to visitors during their stay in a city resort. The service operator is a local tourist agency that, by means of this service, is able to aggregate multimedia content with tourism information into the hybrid broadcast/broadband distribution platform.

A. Description

Tourists access the service through terminals equipped with the appropriate connectivity, i.e. broadband and broadcast access. When users access the service, they receive an electronic tourism guide in their mobile phones, consisted of a list of resources related to tourist locations, events, facilities or amenities in the tourist area. Figure 1 shows a screen of an electronic tourism guide.

Each resource in the tourist guide includes a brief (multimedia) description. Apart from content assets, the platform operator can also include links to third-party services into the resource descriptions, through the mobile telephony network, such as sending message templates, get contact information, access to external links, make automated reservations, buy tickets, or hire a baby-sitter via web services. As the touristic offer also includes private attractions and services (hotels, restaurants or nightlife establishments) local advertisements are also inserted in the guide, thus helping to defray the cost of the service. Thus, users can browse the list of touristic resources and then access an interactive multimedia description of each resource in the list.

The electronic tourism guide is generated by a recommender engine every time a user access the service. The recommender uses the metadata information about each resource to create a personalized list of items for each user, taking into account the user preferences, their history and the experience of other users with similar profiles, as it will be explained in the next sections. For that purpose, the recommender engine communicates with the users' profiles database and with the content management system (CMS) to obtain the assets offered by the tourist agency.

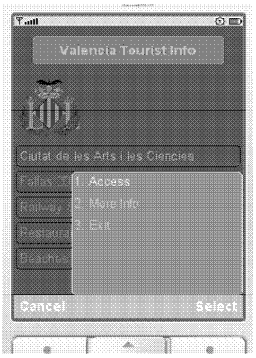


Fig. 1. Electronic Tourism Guide.

But the main characteristic of this guide and the service is that the interactive multimedia descriptions of the resources are not delivered together with the tourist guide. Instead, the platform uses broadcast networks to distribute the most popular elements, while the rest are available through the broadband networks, together with the interactive services. When the user selects one of the resources, the client application fetches the associated description from either the broadcast or the broadband network, according to the access information provided by the tourism guide. This way, the load of the networks is balanced in a transparent way for the user, reducing the congestion and improving the performance of the service. While the user browses the personalized list of resources, the client application joins the broadcast channel in order to cache the resource descriptions transmitted therein. When the user selects a resource it will first check the cache memory and in the event of a cache miss, it will access a web server to fetch the description of that particular service. Figure 2 presents the temporal diagram of the service in a simple manner.

The platform uses IP protocols (namely FLUTE and HTTP) to deliver the data. This way, the service can be offered in a variety of networks, as long as there is support for IP multicast. Currently, there are two kinds of network architectures able to offer the service to commercial mobile terminals. As mentioned in the introduction, mobile TV infrastructure can be used to broadcast the most popular resources. Thus, the service can be offered to mobile phones capable of receiving broadcast mobile TV services. Another alternative is to use IP multicast over Wi-Fi. In this case, it is necessary that the terminal supports IP multicast (i.e. that it implements the IGMP protocol).

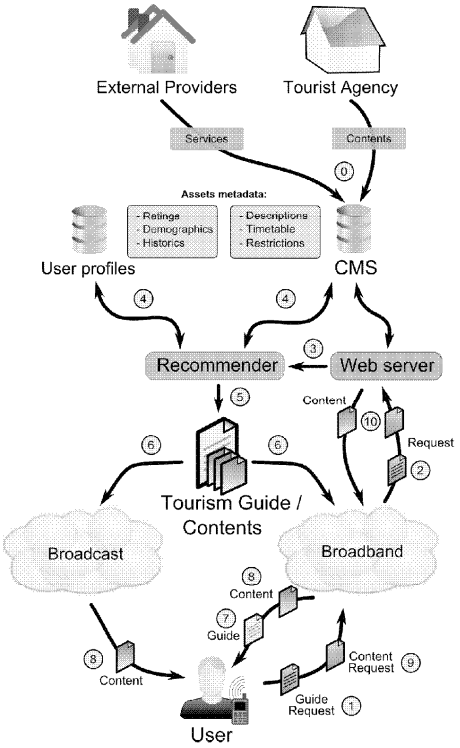


Fig. 2. Diagram of the service prototype.

B. Personalization

The success of the hybrid content distribution scheme promoted in this project is largely dependant on its effectiveness to satisfy users. This is especially difficult in major destinations due to the vast and heterogeneous touristic offering, parallel to the visitors' demographic variety. This way, it is not easy to provide a complete and general offering of touristic resources without overwhelming the users with a countless collection that makes it difficult to take advantage of the service.

Consequently, a recommendation service that suggests resources to visitors depending on their characteristics is a must in this kind of framework where the user terminal is individualized (though it may gather information about the visitors' travelling companion) [1] [2].

To this extent, this project implements a selection mechanism that, taking into account the available metadata for each resource, performs an ordering of touristic resources according to their interest for a given visitor. From this ordering, the recommender suggests a personalized list of resources considered the most appropriate to be included in the guide (see figure 2). That is, the touristic guide is a personalized one, generated by a recommender engine when users access the service, aimed at keeping low the number of offered resources to adapt the service to mobile usage.

This service requires the user to provide some general information to initialize his/her profile (demographic data and personal preferences). This initial profile is built starting from the most similar of a set of predefined prototypical ones, to be later refined to specify particular circumstances of the current visit: companion (kids, disabled…), budget, data constraints, etc. This personal profile is continuously updated with the new information about the visited places, ratings assigned, services contracted through the platform, etc. This way, the platform knowledge about the user gets increased with the usage, improving the accurate of the suggestions.

As an additional personalization feature, the private advertisements about the local attractions and facilities inserted in the guide are also selected to adapt to the user, as well as the coupons and discounts offered by the managers. As a result, only a reduced selection of lures and incentives are offered to each user, avoiding the overwhelming of information and helping to easily handle the available alternatives.

III. Service Implementation

A. Service Architecture

To be able to provide the services previously described, this project implements a framework with the architecture shown in figure 3. There, it is possible to identify four main components or layers: data and content repositories, middleware, web services, and client access applications.

The information stored in the core of the system is basically composed of the multimedia assets and third-party services components to be rendered in the user's terminal via broadcast or broadband networks. These elements and their description metadata are stored and handled by the Content Management System Alfresco[2], a very versatile open-source CMS having extensibility as its main virtue, what makes easy to arrange information in the more convenient way and develop new access interfaces to store and recover it. Completely separated from these, the metadata describing both elements (multimedia contents and external services) are formalized by means of two OWL[3] ontologies that provide the information that drives the personalization processes. The population of these ontologies is completely automated from the metadata introduced by the system administrators in the Alfresco tool, by means of the corresponding XSLT transcoding style-sheets.

The middleware component is an abstract layer that provides format-agnostic access to content and metadata. The most significant technology contained in this Java software is the Jena[4] OWL API used to browse the ontologies, and fetch explicit data or sets of elements bound by some specific relationships.

[2] http://www.alfresco.com
[3] http://www.w3.org/TR/2009/REC-owl2-overview-2009-10-27
[4] http://jena.sourceforge.net

The platform provides external access to the above repositories of data (through the middleware) via a set of web services. These web services are accessed over the broadband mobile network (for client devices) or HTTP Internet connections (for system managers and third-party service providers). Two rows of web services can be observed in the figure 3: in the bottom line, those listening system administration requests (to add/remove/manage content/services/profiles, to select the broadcast assets or the services offered…), and, in the top line, those serving user's requests and service management. These web services have been developed following a REST architectural style that maximizes reusing, minimizes coupling and favors evolution.

Finally, those web services are accessed by client applications to provide real functions to every actor of the system. For example, system administrators (at the bottom of the figure) use a set of Internet applications to maintain the information and to manage the provision of the service. Those applications have been developed as web pages (with AJAX and JavaScript) and call the functions provided by the bottom raw of web services by means of HTTP requests through Internet. The top raw of web services, on the contrary, are mostly devoted to serve the user's mobile device requests: fetching the structure of the tourist guide or some of its contents, changes to the personal profile or calls to third-party services. The client applications running on the user's device that use those web services have been developed in Java, as J2ME MIDP2 applications, making use of the KXML2 API (http://kxml.sourceforge.net) to parser configuration and template files. KXML2 has been thought for MIDP-alike applications from the beginning and implements a pull parser that uses the event mode of SAX but keeping the control of the parsing in the application side, which must request parsing events in a proactive way.

Last, some of the web services of the top line are also devoted to serve the service providers' requests, for instance to register, deploy and control the mobility services associated to the touristic resources. The access to these services by the users is always initiated through the central platform, although they continue in a peer-to-peer way. This permits the platform to annotate the user's activities to improve recommendations and establish a pay-for-click strategy to charge to the service providers for their use of the platform.

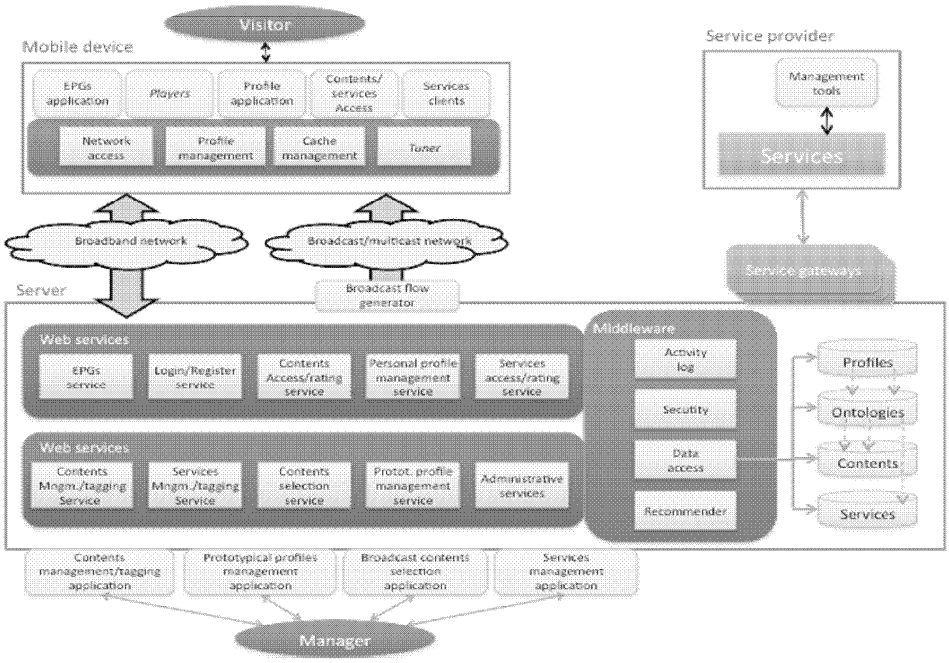


Fig. 3. System architecture.

B. *Content Distribution to Mobile Devices*

This service regards the usage of broadcast access as a mean to push multimedia content to mobile devices. Our service assigns the delivery of the most popular content items to the broadcast access network. Hence, the broadcast unidirectional channel relieves the traffic load caused on the mobile broadband network when users access these items.

There are two different strategies to reach many simultaneous mobile users, mobile broadcast and mobile multicast. In this paper, we generally refer to both as broadcast. The first, strictly broadcast, refers to the ubiquitous transmission of content to all users in a service area. In the second, interactive multicast, the network starts sending multicast data upon request from the first subscriber in the service area. Successive subscribers join the same multicast connection, and the transmission stops when the last subscriber leaves the multicast channel.

Many digital television broadcast systems have developed amendments to provide mobile IP broadcast services within their broadcast networks, DVB-H, DVB-SH or ATSC-M/H among these. Other relevant mobile broadcast technologies are Ofcom's MediaFlo and 3GPP´s IMB. This consortium has also elaborated the MBMS specification to support interactive multicast over 3G cellular networks. These networks use dedicated spectrum resources and consequently independent infrastructure. The 802 family of standards also regards the use of IP multicast on top of wireless technologies such as Wi-Fi and WiMAX. In this case, the multicast data may share the wireless channel with unicast point-to-point connection.

Mobile broadcast content distribution requires of a reliable transmission mechanism that performs well in the presence of losses in the wireless transmission channel. FLUTE (File Delivery over Unidirectional Transport) is a protocol for the delivery of files over IP multicast networks [3].

FLUTE file transmissions are organized in delivery sessions. Each session is uniquely identified by the IP multicast (class D) destination group and a Transport Session Identifier (TSI). Furthermore, each session consists of one or several channels, each one using a different UDP port. Client discovers the session parameters through off-line mechanisms outside the scope of the FLUTE RFC (e.g. SDP). Within the file delivery session, clients discover the number of files and their metadata descriptions by parsing the XML defined File Delivery Table (FDT). Thus, the FDT announces the session contents and the first step taken by a client that joins a FLUTE session is normally to parse this object.

Two mechanisms provide reliability to FLUTE file delivery sessions, namely dynamic file carrousels and Forward Error Correction (FEC) coding. The former consists of the continuous transmission of files in the form of dynamic carrousels, whereby each file is transmitted cyclically with a certain cycle time. Files can be updated and added or removed from the carrousel at any given time.

Regarding FEC, FLUTE adopts the FEC Building Block [4] to inform receivers about the transmission of redundant (parity) data, which helps receivers to recover data lost in the transmission over the erasure channel. The RFC allows the use of different FEC codification algorithms, such as Reed Solomon (RS) coding, Low Density Parity Check (LDPC) or Raptors codes. To apply any FEC encoding, the file is first divided in blocks, which are the FEC encoding / decoding unit. Each source block is later divided in k source symbols, with the constraint that each FLUTE packet must transport an integer number of symbols. The FEC algorithm generates n encoding symbols from every source block. Thus, the FEC encoding rate is k/n, being typical ratios between 2/3 and 2/5.

This way, in the presence of losses, receivers may be able to recover the k symbols of a source block even if some of the source symbols are lost. In case receivers are not able to recover the block from the received data, they must wait to the next file transmission cycle.

In hybrid networks, it is possible to associate a repair service to the file delivery session, so that receivers can obtain missing data through a unicast connection. Depending on the implementation of the file repair service, receivers can request missing symbols, blocks or the complete file.

C. *Personalization framework*

Regarding the recommendation system, a personalization engine has been developed in the project that is in charge of assigning priority to resources depending on the visitors' characteristics. This engine is an adaptation of the well-known recommender AVATAR [2], tested in different environments such as TV content recommendations or personalized composition of distance learning courses over TV.

In this tourism context, the recommendation software of the AVATAR project has been adapted to estimate the adequacy of a touristic resource for a visitor, according to the available information composed of:

- The characteristics of the touristic resource, taking into account the seasonal constraints.
- Demographic data about the visitors and their preferences.
- Their explicit ratings for previously visited resources. This includes resources located in other destinations as long as the underlying recommendation platform is shared.
- Implicit conclusions about their preferences, extracted from their historical behaviour regarding services contracted through the platform (tickets, restaurants, reservations…).

The recommendation service implements a hybrid strategy to estimate the interest of a target user in a touristic resource. That strategy is composed of two chained stages: a content-based approach that compares the characteristics of both the touristic resource and the user, searching similarities between them; and a collaborative filtering analysis that identifies the more similar users to the target one (called neighbors), to later estimate the interest of the target user in the resource from the satisfaction of those neighbors who have rated it.

Following the general approach of the AVATAR project, the kernel algorithm that implements the aforementioned strategies is enhanced with new techniques extracted from the Semantic Web that permit to discover hidden significant relationships between the actors of the framework and the available contents and services. To this aim, all the

information describing the touristic resources must be organized by means of an ontology that classifies hierarchically the touristic resources and provides the significant properties to characterize them and establish the links with other resources and with the user's profile data.

This enhanced strategy explores and weights relationships derived from the properties that characterize the resources in the ontology, searching for semantic links that relate the local tourist resources to those ones positively rated by the user in the past or inferred to be appealing to the user by the system.

In this project, we have developed an OWL ontology named R-OnTour, starting from the well-known OnTour ontology created in the OnTour project [5] and extending it with new classes and properties to include and organize the relevant data that feed the reasoning process that discover new information.

As suggested by figure 4, the ontology permits to declare all the relevant touristic resources of a destination, together with their activities, infrastructure description, access data (location and timetables) and associated multimedia assets to be sent to the user's terminal (videos, images, audio), taking into account device characteristics as can be the screen size, resolution, etc.

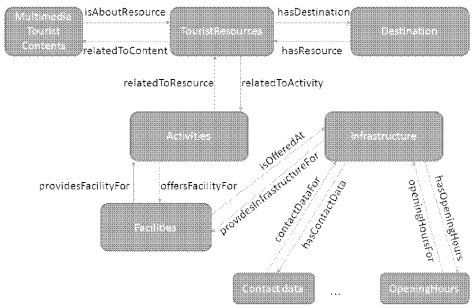


Fig. 4. Excerpt of the R-OnTour hierarchical structure.

The original OnTour is a test-case ontology, developed to demonstrate the capabilities of this technology in the tourism environment. For that reason, it is basically centered in describing accommodations, the nearby activities and the infrastructure where they are developed. That is, it is focused in travel planning, where accommodation plays a center role.

On the contrary, our project is aimed at providing rich support to visitors in situ, making suggestions to visit places once you are in the destination. This way, it is the concept of *TouristicResource* and not the *Accommodation* one which plays the main role. In the same line, the other principal actor in the hierarchy is the concept of *MultimediaTouristContent* that represents a multimedia asset to be sent to the user's device. Additionally, the concept of *Facility* is also powered in our data structure, which tries to gather very important things in this kind of unplanned visits, as can be transport, restaurants, reservation services, commerce…

Finally, as personalization is a main feature of the result, the ontology has been enhanced with a class to store the profiles of the users: personal data, ratings, previous visits or services contracted… This approach, involving a hierarchical description of the user profile, is one of the main characteristics of the AVATAR approach and the cornerstone of its reasoning capabilities, as described in [2].

Of course, the original support for description provided by the OnTour ontology is also used, as can be the *Activities* class. But, once again, the different conception of both projects is reflected in the result. While the OnTour project continuously points to stable activities, usually associated to infrastructure, we pay special attention to irregular events that can go unnoticed to the user: gastronomical fairs, animal exhibitions, sportive events, etc… (after all, nobody needs to be recommended the Eiffel Tower when visiting Paris).

D. Latency

In order to provide a satisfactory experience, it is also important to handle the latency of the file delivery over the hybrid distribution network.

Looking at the broadband access, the average latency is mainly dependant of the network delay in overload conditions. To prevent the occurrence of overload events, a back off timing mechanism that distributes the user requests over time is introduced. Most popular files will need larger back off timing windows to level out the access peaks, thus experiencing longer access times.

Hence, the placement of popular files in the broadcast access helps at keeping the service latency under control, especially in tourist areas where a high density of users per cell is expected. As explained in [6], the average access time to files in the broadcast service area can be optimized, given that an empiric measure of the popularity of files is available. In our implementation, the recommender subsystem provides such ordering of multimedia assets according the estimated interest of the users currently registered in the area, what helps to decide which files must compose the broadcast carousel and their respective cycle time. This way, files with higher ratings have a lower cycle time. Similarly, the back off timing parameters of the files in the broadband access are also given by their respective ratings, as an estimate of their popularity, to minimize the probability of network overload.

IV. Implementation Details

A. Guide Format

One of the requirements accounted for in the platform design is that the service should be offered as an interactive service associated to a mobile TV channel. To accomplish this, our implementation adopts the OMA BCAST Mobile Broadcast Services specifications [7]. In OMA BCAST, the Electronic Service Guide (ESG) contains all the metadata that describes the mobile TV offer. A mobile TV service can include interactive services by referencing an XML file, the *InteractivityMediaDocument* (IMD), which is not transmitted together with the ESG metadata. Instead, the ESG provides access information to instances of IMD documents, which can be delivered through broadcast or interactive mechanisms.

The IMD document allows the presentation of interactive components of a mobile TV service. Instances of IMD documents consist of one or several *MediaObjectGroups* (MOGs) elements and an *ActionDescription* element. The first MOG in the IMD document is presented when the user starts the interactive component of the mobile TV service. The purpose of the *ActionDescription* element is to enable time-dependent behavior of interactive components, by defining which MOG should be presented when the user interacts with the service or which one should be presented after a period of time without interaction. Each MOG consists of *MediaObjectSet* (MOS) elements, which bundle related media objects meant to be rendered together (e.g. an XHTML page and the pictures on it). From each media object group, only one media object set is rendered at the same time by the terminal. This is the MOS with the highest value of the attribute 'RelativePreference'. Lastly, the media objects of a MOS are packed into a GZIP file and the attribute 'Content-Location' of the MOS element tells terminals where to find it.

In our implementation, the recommender uses the semantics of the IMD Schema and the web services defined by OMA for the interactive delivery of IMDs instances. Thus, the tourist guide is an XML document conformant with the IMD document model, where each MOS corresponds to a tourist resource selected by the platform for that particular user. Likewise, tourist resources are GZIP files containing the media items and an XML file with presentation information.

The client application uses the URI provided in the 'Content-Location' attribute of every MOS to fetch the GZIP files of every resource in the list. The access information can refer to either a HTTP (unicast) or FLUTE (multicast) access location. Figure 5 shows an example of an IMD instance, where these two types of access are present. A broadcast cache manager receives the FLUTE URLs and prefetches the selected broadcast contents into the terminal´s memory.

```
<InteractivityMediaDocument id="79588" groupID="5461" groupPosition="1"
version="1">
 <MediaObjectGroup id="0" startMediaFlag="true">
   <MediaObjectSet relativePreference="0" Content-Type="application/x-gzip"
   ContentLocation="http://comm.iteam.upv.es/RH/restaurants.gz">
         < Description> Restaurants </Description>
   </MediaObjectSet>
   <MediaObjectSet relativePreference="0" Content-Type="application/x-gzip"
   ContentLocation="flute://224.12.12.10:80/1/beaches.gz">
         < Description> Beaches </Description>
   </MediaObjectSet>
 </MediaObjectGroup>
</InteractivityMediaDocument>
```

Fig. 5. Interactivity Media Document example.

The implementation of the HTTP and FLUTE protocols for file delivery complies with the File Distribution Service specification of the OMA BCAST specifications. The platform uses the web services defined in [8] for the distribution of files over backend interfaces.

Therefore, the service can be offered as an interactive component of a mobile TV service on clients compatible with the OMA BCAST specifications or as a standalone service. There is a difference between these two use cases: in OMA BCAST terminals, the IMD document triggers the terminals to render the contents of one MOS onto the GUI, whereas our client application presents a browsable list of services.

B. Laboratory Tests

The service under study has been prototyped and tested in different laboratory tests. The tourist agencies involved in the project have provided tourist mobile content multimedia (text, audio and video items) from two different locations (Oinati in the Basque Country and Santiago in Galicia). Once the resources have been created, they are entered in the CMS Alfresco, and characterized according to the ontology used by the recommender. User profiles are managed in a similar way.

The FLUTE file download service has been implemented specifically for this project, since the access to the FLUTE system library in available DVB-H mobiles is not allowed. The service supports the creation and management of dynamic carrousel sessions and the use of different channels per session. It is fully compatible with standard specifications [7] [9] and it has been tested against other FLUTE implementations (including the native ESG client in Nokia's DVB-H terminals). In order to add files to FLUTE sessions, the platform uses the web services that have been defined specifically for this purpose.

On the client side, the prototype client application has been developed for the J2ME MIDP platform. The client application connects to the recommendation web service in order to obtain the IMD document with the personalized electronic tourism guide. The client application parses the IMD document and renders a list of the services onto the GUI. The IMD parser also resolves the access information provided in the IMD document and provides a broadcast cache manager with the FLUTE URLs. Then, the broadcast cache manager uses a MIDP FLUTE library to prefetch the broadcast contents into Record Management Store (RMS) memory. By means of RMS, MIDlets can store data and retrieve it later.

The reference terminal is Nokia´s N96 multimedia smart phone, with both Wi-Fi and DVB-H connectivity. Platform and client application have been evaluated in two different laboratory test scenarios, described hereafter.

1) DVB-H Laboratory tests

A DVB-H multimedia broadcast covers wide outdoor areas. In our service, DVB-H is used to broadcast the most popular content items. The case under study is the use of the service as an interactive component of a mobile TV service. Thus, FLUTE download sessions are bound to mobile TV sessions. There are two different ways to access the service in OMA BCAST enabled terminals: through the native OMA BCAST ESG client and through a standalone MIDP2 application. The standalone J2ME application is running on background when the user is enjoying the Mobile TV service, as depicted in figure 6.

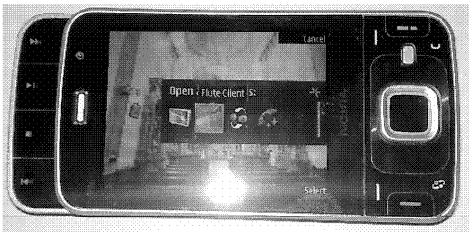


Fig 6. Access of client application with mobile TV service on background

In the laboratory tests, a Cellmetric Modus 3 test modulator is used to broadcast the MPEG-TS encapsulated FLUTE sessions with the ESG, the FLUTE sessions corresponding to the file download service and the RTP sessions delivering mobile TV contents.

2) Wi-Fi Laboratory tests

Wi-Fi is far more extended than DVB-H. It is a good alternative for indoor environments and relatively small coverage areas. Thus, it is a good choice for providing the service in touristic resorts.

In the Wi-Fi laboratory tests, the platform connects to the FLUTE server, which multicasts over a Wi-Fi Access Point. Client applications join this channel to receive multicast content. The multicast traffic over Wi-Fi works at a speed that must be configured in the Access Point, given the application requirements. The multicast coverage range is inversely proportional to the multicast speed.

The client application has been tested in different terminals, such as Nokia's N77, E61, E65 or E90. In the Wi-Fi tests, the FLUTE client application is pushing content into the cache memory whenever the terminal is inside the multicast coverage area of the Access Point.

V. Related Work

The rising of the interest in convergence and cooperation of mobile networks in last years is definitively witnessed by the many projects that can be referenced and compared to this work. As expected, no one of these projects shares the whole set of goals or technologies present in the work described in this paper, but different comparatives can be done according to multiple axis, and the following ones represent only a highlight of these analyses.

INSTINCT [10] is a project of the European VI Framework Program. Originally thought in the early days of the DVB-H deployment, INSTINCT intended to be a general framework for the development of the infrastructure to create and deploy hybrid services as the one described in this paper, involving the collaboration of the broadcast (DVB-T and DVB-H) and broadband (GPRS/UMTS) networks.

Of course, INSTINCT aims to define an open final platform for the delivery of convergent hybrid services, exploring the communication between Java MIDP applications and MHP ones to be able to access (part of) DVB services through mobile devices. But its goals are far more ambitious and generic, ranging from a general system architecture (including DRM) or network equipment (none of them object of our project), to more content-centric issues as the identification of business scenarios, the development of trials and training activities, or the provision of tools to create hybrid contents and services (similar to ours in the sense of providing service creators with a web service set of calls to register external services).

Another related project is AMUSE [11], an Austrian project involving network operators and manufacturers as Alcatel. Its main goal is to develop a testbed for hybrid services, providing access to contents in a bearer-agnostic way with unified interactive/broadcast clients. That testbed should provide infrastructure to develop trials involving real users and a new generation of hybrid interactive services including social networks by mean of P2P and similar technologies. To provide such testbed in a cost-effective way, it defines an optional mechanism to receive the DVB-H stream by means of a laptop (with a richer set of library capabilities to process the information) to later transfer the processed data to the mobile device via Wi-Fi. Such solution is a workaround for early stages where programmable devices were poor and scarce, and it is no longer necessary for state-of-the-art terminals like ours.

Significantly, there is a common concern with our project regarding the necessity of a software agent (Cooperative Management Platform or CMP) to be in charge of coordination of content distribution to both networks, for instance for load balancing issues.

The more important contributions of our project with respect to the previous ones are the use of the OMA BCAST specification [8] to distribute contents, and the personalization support to doing so in a selective way.

Regarding touristic mobile services, it is necessary to mention the COMPASS project [12] as it provides a very similar service to users, but only through the 3G network (that is, it is not a hybrid service). This project provides context-aware recommendations to mobile phones about touristic resources in the nearby locations (the service is aware of the user's location and, so, a maximum distance to the point of interest is a hard requirement).

Those suggestions, as ours, are personalized according to a personal profile that is continuously enhanced with the available feedback. As another similarity, third-party services related to the touristic resources are available through the platform. Both touristic resources and services are described by means of the OWL language to capture their semantics, leading to an intelligent framework to make personalized recommendations.

On the other side, as derived by being limited to the broadband network, the number of users and the amount of multimedia data sent to them (e.g. videos) greatly affect to the service performance.

VI. CONCLUSION

This paper has presented a novel tourist information service provisioned through state-of-the-art hybrid networks, where a local platform at a touristic destination use both broadcast and broadband networks to deliver a local multimedia tourism guide to visitors equipped with appropriate terminals. As the guide is built from data received from both networks, an intelligent management of load balancing (broadcasting the more appealing multimedia contents) greatly improves the performance of the service, especially in very popular and crowded touristic areas.

In addition to multimedia contents describing touristic resources, the guide provides direct access to several third-party services closely related to the touristic resources (contact information, ticket buying, slot reservations, etc.) and to general vacation-related issues (restaurant reservation, hire a baby-sitter, access to medical services…).

To improve usability and avoid overwhelming the users with information, tourist guides are personalized for each user according to their preferences and historics. The recommender engine uses a new OWL ontology of touristic resources to discover significant relationships by means of semantic web techniques, what improves the accurate of suggestions, being these touristic resources or external services. In addition, the recommender information is also used to drive cache decisions, minimizing the latency experienced by users.

The service, completely implemented with standard technologies (Alfresco, OWL, Jena, KXML, JavaScript…) follows the OMA BCAST specification of the Open Mobile Alliance, aimed at standardize the signaling and delivering of interactive services over broadcast mobile networks.

ACKNOWLEDGMENT

The authors would like to thank the rest of the members of the project consortium for their comments and suggestions: Gesfor, Moviquity, VICOMTech, Ericsson, Batura, Goiena and TurGalicia.

REFERENCES

[1] F. Fraile, I. de Fez, J. C. Guerri, "Modela-TV: Service Personalization and Business Model Management for Mobile TV", 7th European Interactive TV Conference (EuroITV), Leuven (Belgium), June 2009.

[2] Y. Blanco, J. Pazos, M. López, A. Gil, M. Ramos, "AVATAR: an improved solution for personalized TV based on semantic inference", *IEEE Transactions on Consumer Electronics,* vol. 52, pp. 223-231, 2006.

[3] T. Paila, M. Luby, R. Lehtonen, V. Roca and R. Walsh, "File Delivery over Unidirectional Transport", RFC 3926, October 2004.

[4] Luby, M., Vicisano, L., Gemmell, J., Rizzo, L., Handley, M., and J. Crowcroft, "Forward Error Correction (FEC) Building Block", RFC 3452, December 2002.

[5] Katharina Siorpaes and Daniel Bachlechner, OnTour: Tourism Information Retrieval based on YARS, 3rd European Semantic Web Conference (ESWC-2006), Budva (Montenegro), 11 – 14 June 2006.

[6] G. Zhiqi, Y. Songyu, Z. Wengjun "Using object multiplex technique in data broadcast on digital CATV channel" *IEEE Transaction on Broadcasting*, Vol. 50, Issue 2, June 2004.

[7] Open Mobile Alliance™, OMA-AD- BCAST-V1_0 "Mobile Broadcast Services Architecture", 12 February 2009.

[8] Open Mobile Alliance™, OMA-AD- BCAST-V1_0 "File and Stream Distribution for Mobile Broadcast Services", 12 February 2009.

[9] ETSI, TS 102 472 v1.2.1: Digital Video Broadcasting (DVB); IP Datacast over DVB-H, Content Delivery Protocols" (2006).

[10] S. Schiek, P. Steckel, "MHP-based mobile prototype implementing the INSTINCT middleware concept", 14th IST Mobile and Wireless Summit, Dresden (Germany), 19-23 June 2005.

[11] R. Schatz, N. Jordan, S. Wagner. "Beyond Broadcast: A Hybrid Testbed for Mobile TV 2.0 Services", 6th Int'l Conference on Networking, Sainte-Luce (Martinique), April 22-28, 2007.

[12] M. van Setten, S. Pokraev, J Koolwaaij, "Context-aware recommendations in the mobile tourist application COMPASS", *Lecture Notes in Computer Science*, Volume 3137, pages 235-244.

BIOGRAPHIES

Alberto Gil was born in Domayo, Spain in 1968. He received his Telecommunications Engineering degree from the University of Vigo, Spain, in 1991, and his Ph.D. degree in Telematics from the same university in 2000. Currently, he is an associate professor in the Department of Telematic Engineering at the University of Vigo, and a member of the Interactive Digital TV Laboratory. He is involved with different aspects of middleware design and interactive multimedia services, mainly related to personalization.

Francisco Fraile (M'09) was born in Murcia, Spain in 1979. He obtained a degree in Telecommunication Engineering from the Polytechnic University of Valencia (UPV) and a M. Sc. Degree in Microwave Engineering from the University of Gävle in 2004. Since then he works as a research engineer for the Swedish company Interactive TV Arena. In 2006, Francisco joined the Multimedia Communications research group (COMM) of the Institute of Telecommunications and Multimedia Applications at UPV, to proceed with his doctoral studies as an industrial Ph.D. student, interested in networked electronic media.

Manuel Ramos was born in Lugo, Spain in 1966. He received his degree in Telecommunications Engineering from Polytechnic University of Madrid, Spain in 1991, and his Ph.D. degree in Telematics from the University of Vigo, Spain in 2000. Since 2001, he is an associate professor in Telematics Engineering at the University of Vigo. His research topics are Interactive Digital TV concentrating on recommender systems, integration with smart home environments and interactive applications design and development.

Ismael de Fez was born in Valencia, Spain in 1983. He received his Telecommunications Engineering degree from the Polytechnic University of Valencia (UPV), Spain, in 2007. His M.Sc. thesis was awarded by the Telefónica chair of UPV. Currently he works as a researcher at Multimedia Communications research group (COMM) of the Institute of Telecommunications and Multimedia Applications (iTEAM). He is taking an M.Sc. Course in Telecommunication Technologies. His areas of interest are file transmission over unidirectional environments and file encoding.

Juan Carlos Guerri was born in Valencia, Spain in 1969. He received his M.S. and Ph. D. (Dr. Ing.) degrees, both in Telecommunication Engineering, from the Polytechnic University of Valencia, in 1993 and 1997, respectively. He is a professor in the E.T.S. Telecommunications Engineering at the Polytechnic University of Valencia, and he leads Multimedia Communications research group (COMM) of the iTEAM (Institute of Telecommunications and Multimedia Applications). He is currently in research and development projects for the application of multimedia to industry, medicine, education and communications.

-------- Mensaje original --------

Asunto:IEEE Transactions on CE - Author Notification (#6805)
Fecha: Tue, 16 Feb 2010 10:06:56 -0500 (EST)
De:w.luplow@ieee.org
Para:agil@det.uvigo.es

Dear Alberto Gil,

This is to inform you that your paper entitled "Personalized Multimedia Touristic Services for hybrid broadcast/broadband mobile receivers" will be published in the February 2010 issue of the IEEE Transactions on Consumer Electronics of which a complimentary copy will be sent to all authors.

Thank you for your interest in the Consumer Electronics Society of the IEEE.

Sincerely,

WAYNE C. LUPLOW
Editor-in-Chief
IEEE Transactions on Consumer Electronics