Aplicaciones Móviles En Entornos Servidor

Roberto Casado-Vara [1]

[1] University of Salamanca, Plaza de los Caídos s/n – 37002 – Salamanca, Spain
rober@usal.es

Resumen: En este documento vamos a analizar desde diferentes facetas la transmisión de datos en los entornos celulares. Empezaremos analizando en profundidad las distintas características de las redes móviles celulares que implicarán consecuencias realmente interesantes de estudiar en la transmisión de datos en estas redes. Posteriormente analizaremos algunos modelos de negocio que se establecen en las aplicacio-nes de descargas de contenidos y de acceso a Internet. En este apartado haremos un análisis en profundidad de los modelos de WAP e i-mode estableciendo semejanzas y diferencias. Finalmente, los dos últimos apartados giran en torno a la parte más tecnológica de las aplicaciones cliente-servidor. La descarga segura de contenidos es un tema importante en el mundo móvil porque ha generado bastante dinero y porque se considera como una de las mayores fuentes de ingresos futuros. Sobre este tema han ido apareciendo tecnologías propietarias que finalmente han desembocado en un estándar común tanto de los procedimientos de descarga como de los métodos de protección de los contenidos. Además, también analizaremos los protocolos de transmisión, las características y los forma-tos de los contenidos multimedia. La transmisión de sonidos, imágenes e incluso vídeos es considerada como una de las grandes oportunidades para aprovechar los grandes anchos de banda que están disponibles en las redes 3G.

Palabras clave: aplicaciones móviles

Abstract. In this document we will analyze from different facets the data transmission in cellular environments. We will begin analyzing in depth the different characteristics of cellular mobile networks that will imply really interesting consequences of studying in the data transmission in these networks. Later we will analyze some business models that are established in the applications of content downloads and Internet access. In this section we will make an in-depth analysis of the WAP and i-mode models, establishing similarities and differences. Finally, the last two sections revolve around the more technological part of client-server applications. Safe downloading of content is an important issue in the mobile world because it has generated a lot of money and because it is considered to be one of the biggest sources of future income. Proprietary technologies have been appearing on this subject, which have finally resulted in a common standard of both downloading procedures and content protection methods. In addition, we will also analyse the transmission protocols, characteristics and forms of multimedia content. The transmission of sounds, images and even videos is considered to be one of the great opportunities to take advantage of the large bandwidths that are available on 3G networks.

Keywords: mobile applications

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241

1. Introduction

Todo el mercado de las telecomunicaciones móviles celulares está sufriendo un importante cambio de mentalidad y objetivos. Hasta hace poco la estrategia giraba en torno a conseguir el mayor número de clientes para los operadores y vender el mayor número de equipos y terminales para los fabricantes. Pero en un sector en el que la penetración de los usuarios ronda el 100%, esas metas han dejado de tener sentido.

Los objetivos ahora son distintos, teniendo un número alto y estable de clientes las estrategias giran en torno a otros fines, sin olvidar por supuesto el aumento de usuarios a costa de la competencia. Si suponemos una base importante de clientes, interesa mejorar la calidad de los mismos (que sean los más posibles de contrato), que gasten lo más posible (aumentar el ARPU, *Average Revenue Per User*) y que cambien lo más posible de terminal.

Para conseguir estos dos últimos ejecutivos es importante definir y desarrollar nuevas tecnologías que permitan por un lado proporcionar nuevos servicios atractivos que den valor añadido al cliente y que permitan mayor número de ingresos por cada usuario, y que por otro lado motivarán que los plazos de cambio de terminal se acorten.

En particular, los operadores están siguiendo claramente una línea de acción con la intención de alejarse lo más posible del modelo de negocio con el que suelen trabajar las *utilities* (energía, agua, etc.) y que se suelen basar en productos sin diferenciar con precios muy ajustados en costes [1-6].

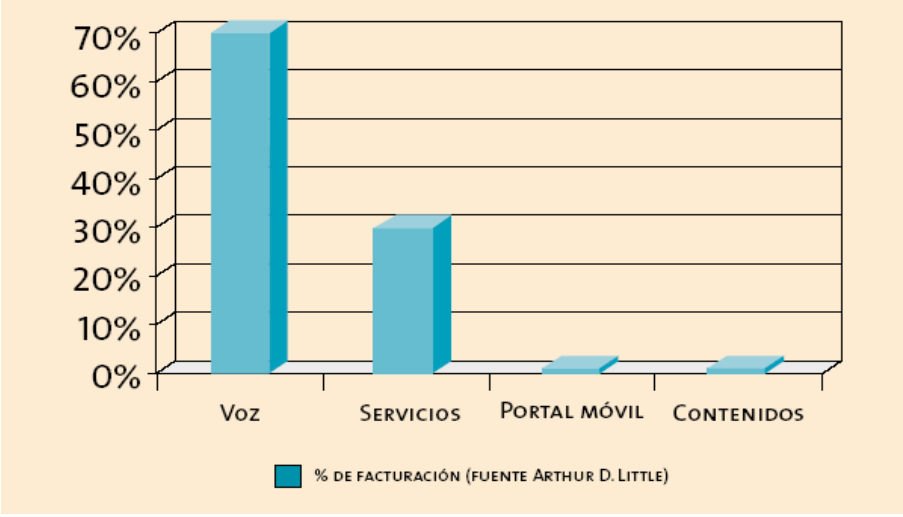

Figura 0.21: Distribución tradicional de la facturación en los operadores móviles

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

Las operadoras móviles actualmente están intentando que los servicios de voz sean parte del conjunto de servicios más amplio que les permita diversificar sus ingresos y aumentarlos sin necesidad de modificar las tarifas o aumentar los usuarios.

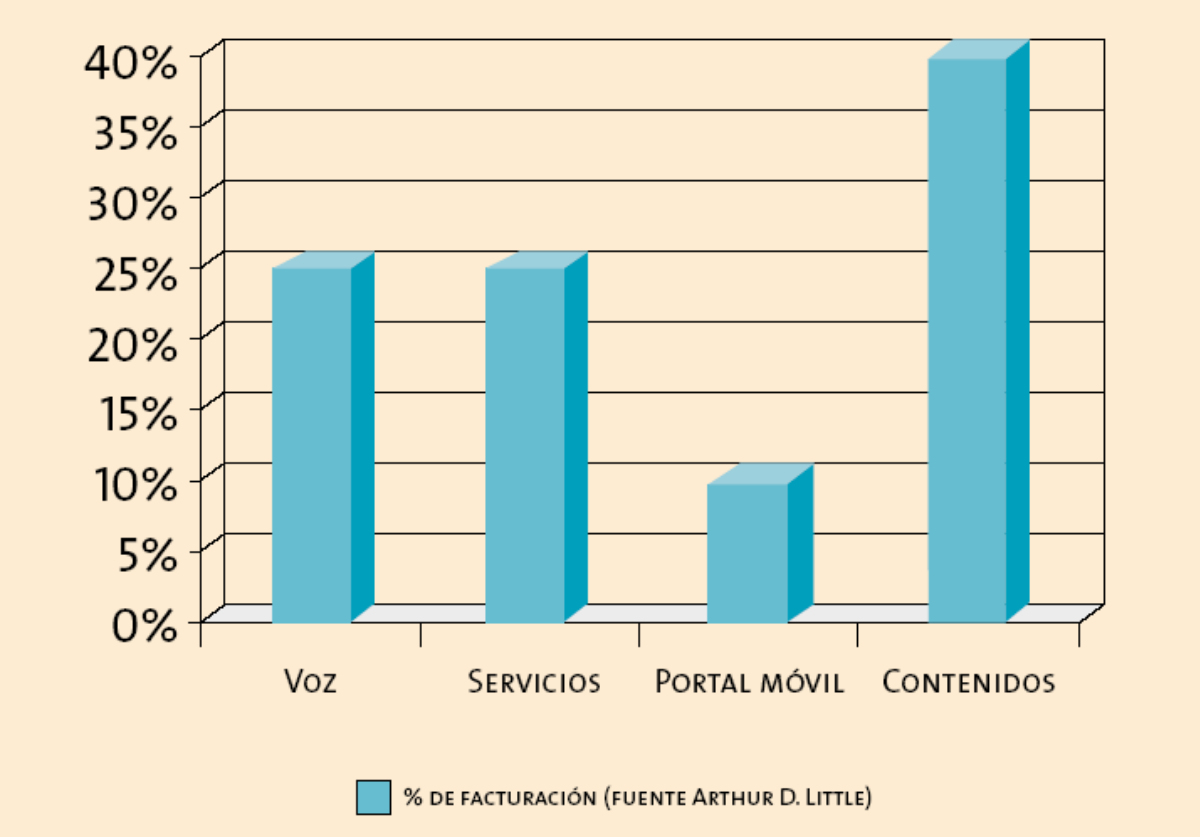

Figura 0.22: Distribución futura de la facturación en los operadores móviles

En este contexto es donde cobra gran importancia todas las tecnologías y modelos de negocio referentes a las aplicaciones cliente-servidor en el mundo móvil. Mediante las aplicaciones que generen tráfico de datos en la red móvil los ingresos aumentarán fácilmente ya sea por el coste del propio tráfico o por el precio de los contenidos o las aplicaciones utilizadas.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

2. TRANSMISIÓN DE INFORMACIÓN EN ENTORNOS MÓVILES

La transmisión de información con redes de telecomunicaciones es un hecho muy habitual en nuestros días. La sociedad asume como normal que cualquier aplicación informática pueda conectarse a otras máquinas (servidores) para obtener o mandar información. Lo que está cambiando poco a poco, es que también los terminales móviles están empezando a necesitar y a promover intercambios de datos más allá de la voz.

Durante los últimos años hemos asistido a un proceso por el cual las redes de comunicación fijas se han ido adaptando al patrón de intercambio basado en datos. Las aplicaciones han ido apareciendo y las redes han evolucionado para proporcionar una velocidad y unas características óptimas para ellas.

En cambio, en las redes móviles estamos en fases anteriores. Hasta ahora la aplicación estrella ha sido la voz y para ella estaban pensadas todas las redes celulares. La única excepción a este hecho ha sido la mensajería a través de mensajes cortos de texto. Aunque incluida en el estándar GSM, no se había previsto ningún tipo de éxito a esta funcionalidad por lo que las redes no estaban preparadas para soportar el tráfico actual. Es el primer ejemplo de cómo las redes han tenido que ir evolucionando tanto en estándares como en planificación para dar soporte a aplicaciónes con intercambio de datos [7-10].

Además de esta esperada evolución, la comunicación inalámbrica tiene intrínsecamente ciertas peculiaridades que afectan directamente a la transmisión de información a través de ellas. Por ejemplo, existen fenómenos como los desvanecimientos o el multitrayecto que no suceden en entornos basados en cables y que provocan que aplicaciones en las que se necesita un caudal constante de datos haya que tener en cuenta que éste se puede cortar y haga necesarios mecanismos para solventar estos problemas.

Aún con todo lo contado, no todo lo relativo a las tecnologías móviles es negativo. En el tema de la seguridad de la información, aunque se utilice un medio compartido y accesible como los enlaces radio, las tecnologías digitales han permitido la transmisión segura tanto de voz como de datos.

En este apartado vamos a desglosar todas las características importantes de la transmisión de datos en los entornos móviles celulares intentando identificar que puntos van a mejorar o a cambiar en las nuevas redes 3G. Además, se va a analizar las consecuencias de estas peculiaridades en el diseño y desarrollo de aplicaciones móviles [11-17].

2.1.Características particulares de la comunicación móvil

Vamos a analizar una por una las principales características de la transmisión de información mediante redes móviles celulares, distinguiendo claramente de qué tipo de redes estamos hablando, (1G, 2G, 2,5G o 3G). En algunos casos, para facilitar la comprensión de esas peculiaridades, vamos también a establecer comparaciones entre las generaciones.

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

Gran parte de estas propiedades son consecuencia de la orientación de las redes a la comunicación de voz y la antigüedad de la mayoría de estándares (GSM data sus comienzos en 1982).

2.1.1 Uso de un medio compartido

Por definición las comunicaciones móviles utilizan el aire para transmitir la información. Evidentemente, este medio es accesible por cualquier agente con el equipo adecuado. Esto podría suponer un problema grave de seguridad en las comunicaciones de datos (y por supuesto también de voz) ya que cualquier persona que se situase lo bastante cerca de nosotros podría "pinchar" la línea.

De hecho, en los primeros sistemas analógicos 1G de telefonía celular escuchar una conversación era tan sencillo como utilizar un analizador de espectro y sintonizar la frecuencia de la conversación. Con la aparición de los sistemas celulares digitales la situación dio un giro de 360 grados. Las redes móviles celulares pasaron de tener una seguridad inexistente o ridícula a tener un sistema prácticamente inexpugnable. A día de hoy existen mecanismos para romper la seguridad de conversaciones GSM pero son realmente complejos y por lo tanto extremadamente caros. Y aún así no se puede realizar en tiempo real. La seguridad de una transmisión móvil con un sistema digital se puede considerar por ahora como muy segura [18-21].

2.1.2 Conmutación de circuitos vs. conmutación de paquetes

Las primeras redes móviles de comunicación estaban orientadas a la comunicación de voz. En esa época, los datos todavía no tenían la importancia actual y era considerados como contenidos alternativos. Eso provocó que hasta las tecnologías 2G, como GSM, estuvieran orientadas a la transmisión de voz mediante canales reservados. Este tipo de comunicación se denomina conmutación de circuitos y se basa en la reserva de un cierto ancho de banda para la comunicación. Por lo tanto en este esquema se utiliza un canal de la misma capacidad siempre se use parcialmente o totalmente. Dependiendo del tipo de tráfico, esto provoca ineficiencias en el uso del sistema y que el usuario tenga que pagar por el tiempo de conexión y no por el uso real que ha hecho.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

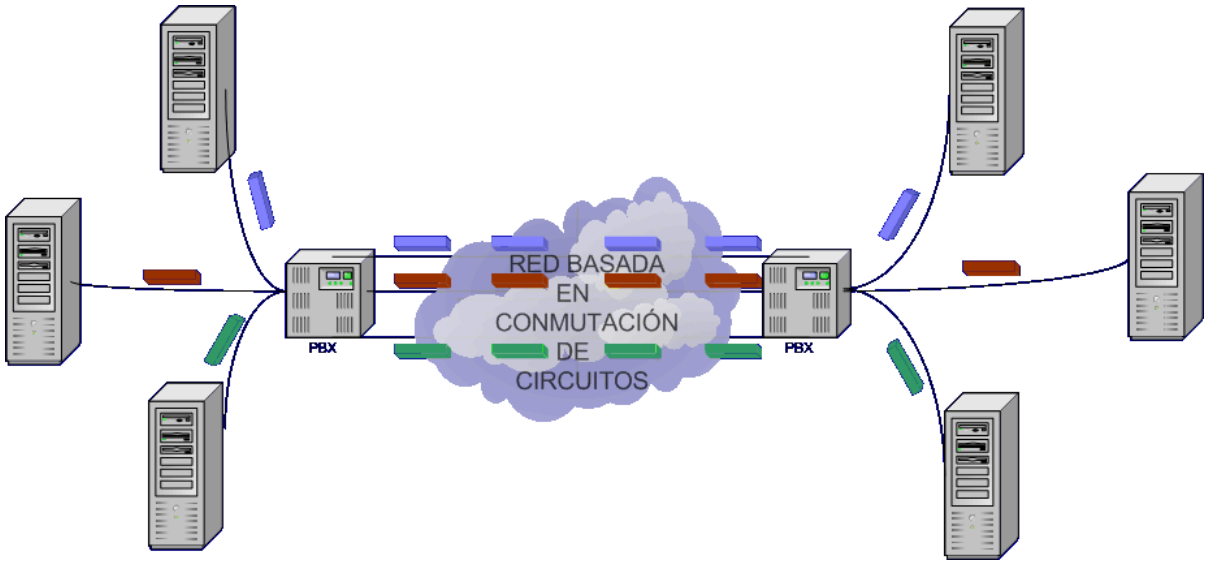

Figura 0.23: Red basada en conmutación de circuitos

A partir de la aparición de las redes 2,5G y 3G, como GPRS y UMTS las redes permiten el uso de conmutación de paquetes en la transmisión de datos. De esta forma se comparten los canales de transmisión de datos entre los diferentes usuarios optimizando el uso de la red. Como consecuencia los usuarios sólo pagan por la información que realmente transmiten en ambos sentidos.

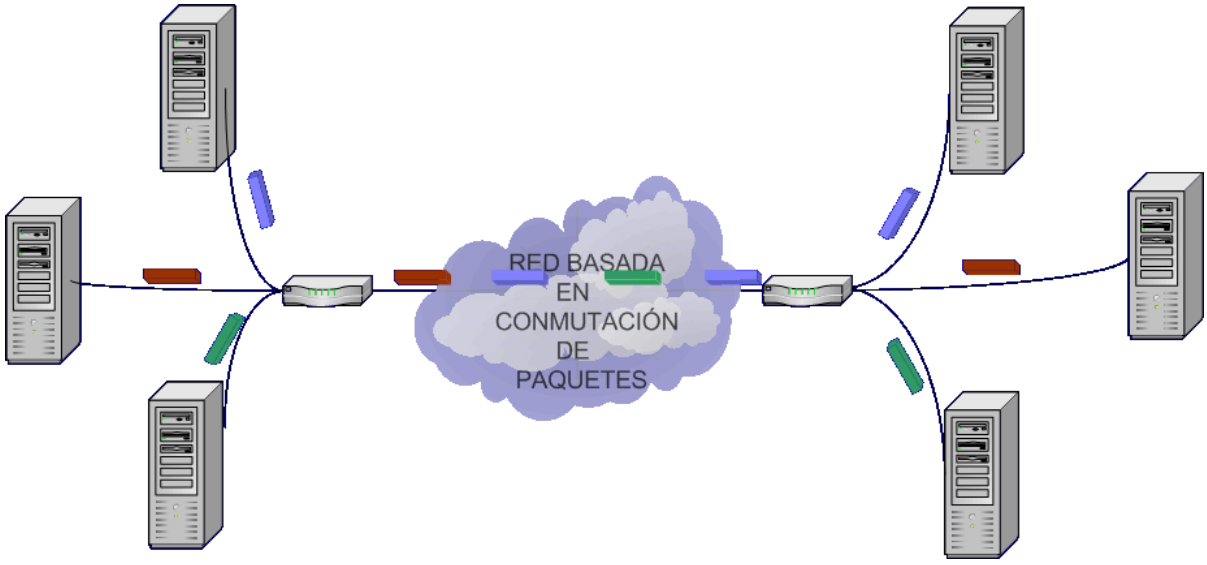

Figura 0.24: Red basada en conmutación de paquetes

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

Es de destacar, que aunque estas características diferencian a las redes móviles de diferentes generaciones, no son peculiaridades de las redes inalámbricas. En las redes fijas existen los mismos esquemas y se producen los mismos problemas y diferencias.

2.1.3 Desvanecimientos de la comunicación

Lo que sí que es una característica especial de las redes móviles son los desvanecimientos de la conexión debido a problemas en la propagación de la señal por el aire. Evidentemente, fluctuaciones en la capacidad de la transmisión se pueden encontrar en cualquier comunicación con compartición del canal (conmutación de paquetes), pero en las redes fijas o con cables, si se reserva un canal para una conmutación, éste va a mantenerse.

En las redes móviles, incluso con conmutación de circuitos podemos encontrar desvanecimientos de la comunicación debido a problemas como falta de cobertura, problemas con los traspasos, efectos negativos del multitrayecto, etc.

2.1.4 Ancho de banda escaso

Otra característica fundamental de las comunicaciones móviles es la limitación que supone el espectro radioeléctrico. Puesto que este recurso es escaso y regulado por la administración competente, los operadores no pueden ampliar las frecuencias a usar de forma arbitraria. Esto se traduce en un ancho de banda limitado y por lo tanto caro. Hasta ahora, cualquier de comunicación de datos mediante redes móviles tenía menor velocidad y un precio comparativamente mucho mayor que la equivalente en las redes fijas.

2.1.5 Retardo alto

Otro problema que tienen actualmente las redes móviles es el retardo elevado que se produce en el establecimiento de conexiones. Este problema cada vez va siendo menor con las mejoras que en este aspecto están introduciendo las redes de tercera generación.

2.1.6 Simetría en las comunicaciones

Otra consecuencia de la orientación de las redes 1G y 2G hacia las comunicaciones de voz es la simetría de la capacidad en la transmisión. El ancho de banda que se proveía en las transmisiones (tanto de voz como de datos) era igual tanto en el enlace ascendente como en el descendente. Esto era claramente un desperdicio según el patrón de comunicaciones de la mayoría de aplicaciones de Internet, en la que el gasto de ancho de banda en el canal descendente es unas diez veces mayor que el correspondiente del descendente [22-25].

Con la aparición de las redes 2,5 esto se ha solucionado en parte puesto que se permite usar más canales compartidos de descarga que de subida. En algunas tecnologías 3G diseñadas especialmente para la transmisión de datos la asimetría es totalmente configurable .

2.1.7 Limitaciones en los terminales

En las redes fijas, sobre todo conectadas a terminales inteligentes como ordenadores, la capacidad de los terminales era más que suficiente para gestionar todos los contenidos y servicios que se podían implementar y transmitir mediante las redes.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

En las redes móviles, los terminales han ido evolucionando a la par que las redes, pero en muchos casos éstas han ido por delante de los dispositivos. Esto ha provocado que muchas veces lo que impedía nuevos servicios y contenidos más avanzados hayan sido los propios terminales y no las redes.

Las limitaciones más importantes de los dispositivos móviles, son la capacidad de conmutación, la calidad de la reproducción de los contenidos multimedia como gráficos y sonidos, las pantallas y por supuesto la interfaz de usuario con teclados numéricos. Estas limitaciones vienen impuestas por la necesidad de que los terminales sean portátiles, con lo que la autonomía, el tamaño y el peso son requisitos imprescindibles.

Este es uno de los apartados en los que la brecha que existe entre las redes fijas y móviles ha sido más grande en tiempos pasados. También es cierto que la diferencia que se está haciendo más pequeña cada vez y que los terminales de dentro de pocos años van a ser pequeños ordenadores portátiles con conexiones a redes celulares, con lo que las diferencias en este aspecto van a ser mínimas e irrelevantes en un plazo corto de tiempo.

2.1.8 Gran variedad de terminales

Uno de los aspectos más característicos de las redes móviles es la gran variedad de fabricantes de terminales y modelos que existen en el mercado. Además las diferencias entre modelos pueden ser realmente importantes. Si comparamos un dispositivo de hace dos años con uno actual, las diferencias entre las respectivas capacidades son sencillamente abismales. También se producen grandes contrastes entre terminales contemporáneos de gama alta y gama baja.

Esta gran variedad de modelos y capacidades introduce una gran complejidad en el desarrollo de aplicaciones y dificulta el óptimo aprovechamiento de las posibilidades de los modelos.

3.2 Consecuencias en el desarrollo de aplicaciones cliente-servidor

En este apartado vamos a tratar de analizar las características de la transmisión de datos en redes móviles desde el punto de vista del desarrollo de aplicaciones en las que se intercambia información con servidores. Así, vamos a tratar de ver las consecuencias y los efectos que producen estas propiedades vistas en el apartado anterior a la hora de transmitir información de todo tipo entre el terminal y el servidor correspondiente [26-30].

2.2.1 Consideraciones de eficiencia en el uso del canal

Como ya hemos visto el espectro y por lo tanto el ancho de banda es un recurso escaso y caro. Por lo tanto en todas las transmisiones que realicemos sobre un canal móvil deben de estar lo más optimizadas posible. De esa forma el usuario tendrá que pagar menos por usar nuestra aplicación, el operador gastará menos recursos de la red para una misma aplicación y la aplicación tendrá unos tiempos de respuesta mejores.

Esto son varios casos que a modo de ejemplo ilustran las diferentes decisiones de diseño que ayudarán a que nuestra aplicación sea eficiente en el uso del ancho disponible:

- En caso de querer transmitir datos textuales entre aplicaciones, hay que evitar utilizar XML. Primero porque su lectura gastará bastante CPU en el terminal, y segundo y más importante

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

porque mete una sobrecarga no necesaria. Hay que buscar formatos sencillos pero que a la vez ocupen el menor espacio posible. En caso de datos muy complejos, podría merecer utilizar XML, pero siempre sería conveniente que las etiquetas fueran realmente cortas. En los siguientes cuadros se puede ver la misma información codificada de diferentes formas más o menos óptimas en su transmisión.

```
<?xml version="1.0" encod-
ing="UTF-8"?>
<ChessGame>
      <PlayerOneName>John</Play-
erOneName>
      <PlayerTwoName>Peter</Play-
erTwoName>
      <BoardOfTheGame>BXBABBZBBBB
BBPRRBBBBBBBBBBBBBB</BoardOfTheG
ame>
      <MoveNumber>23</MoveNumber>
      <GameTime>12:34</GameTime>
      <Turn>Player1</Turn>
</ChessGame>
```

```
<?xml version="1.0" encod-
ing="UTF-8"?>
<CG>
      <p1>John</p1>
      <p2>Peter</p2>
      <b>BXBABBZBBBBBBPRRBBBBB
BBBBBBBBBB</b>
      <m>23</m>
      <g>12:34</g>
      <t>Player1</t>
</CG>
```

```
John;Peter;BXBABBZBBBBBBPRRBBBBBBBBBBBBBB;23;12:34;Player1
```

- Si estamos en una aplicación basada en WAP, hay que cuidar mucho las páginas WML. Una de las funcionalidades más interesantes para minimizar los datos transmitidos son los datos almacenados en las variables WML del navegador, tal y como se puede ver en el siguiente ejemplo.

Figura 1 <?xml version="1.0" encoding="ISO-8859-1"?>
Figura 2 <!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
Figura 3 <wml>
Figura 4 <template>
Figura 5 <do type="options" label="Menu">
Figura 6 <go href="index.jsp"/>
Figura 7 </do>
Figura 8 <do type="prev" label="Atras">
Figura 9 <prev/>
Figura 10 </do>
Figura 11 </template>
Figura 12 <card title="Compra">
Figura 13 <p>Escribe finalmente para la compra del $(producto) tu número de tarjeta
Figura 14 <input name="tarjeta"/>
Figura 15
Figura 16 y la caducidad de la misma

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

```
Figura 17          <input name="mes"/>/<input name="anyo"/>
Figura 18          <a href="#c" title="Comprar">Comprar</a>
Figura 19        </p>
Figura 20      </card>
Figura 21      <card title="Confirmacion">
Figura 22        <p>¿Estas seguro de la compra?<br/>
Figura 23          <anchor title="si">Sí<go href="comprar.jsp">
Figura 24            <postfield name="tarjeta" value="$(tarjeta)"/>
Figura 25            <postfield name="numero" value="$(numero)"/>
Figura 26            <postfield name="productoID" value="$(productoID)"/>
Figura 27            <postfield name="mes" value="$(mes)"/>
Figura 28            <postfield name="anyo" value="$(anyo)"/>
Figura 29          </go>
Figura 30          </anchor>
Figura 31          <anchor title="no">No<prev/>
Figura 32          </anchor>
Figura 33        </p>
Figura 34      </card>
Figura 35    </wml>
```

Otras buenas normas de uso es utilizar los elementos *template* para agrupar los botones y utilizar nombres cortos en todo tipo de variables y parámetros.

Al final, la opción más eficiente para reducir la cantidad de datos transmitidos es usar si es posible una codificación del XML. En este sentido se ha definido el estándar WBXML (*Wireless Binary XML*) que permite reducir la sobrecarga que introduce XML en las comunicaciones móviles siendo usado en WML y WMLScript.

2.2.2 Consideraciones sobre la calidad de la transmisión

Ya hemos visto que el ancho de banda en una red móvil puede sufrir desvanecimientos, sobre todo si el usuario está en movimiento. Estas caídas en la calidad de la transmisión afectan principalmente a las aplicaciones “en tiempo real”, como por ejemplo juegos multijugador o *streaming* de contenidos multimedia.

Las únicas medidas posibles para solventar este tipo de comportamientos son crear *buffers*, almacenes de datos intermedios de los que tirar en caso que la comunicación se resienta. Evidentemente, esto no siempre es posible puesto muchas veces los datos se tienen que entregar y procesar en el mismo instante en el que se generan.

2.2.3 Consideraciones sobre conmutación de circuitos y conmutación de paquetes

En la actualidad, los datos transmitidos cada vez van más por redes con conmutación de paquetes y esta tendencia va a seguir imparable en los próximos años. Aunque conmutación de paquetes tiene grandes ventajas como compartición de recursos, mayores velocidades y facturación por tráfico, no siempre es la mejor opción para todas las aplicaciones

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

En el siguiente gráfico podemos ver cómo se puede comportar el ancho de banda de una transmisión móvil por conmutación de circuitos y por conmutación de paquetes. La clave reside en que mientras que en una de ellas el canal de datos está reservado en la otra se comparte.

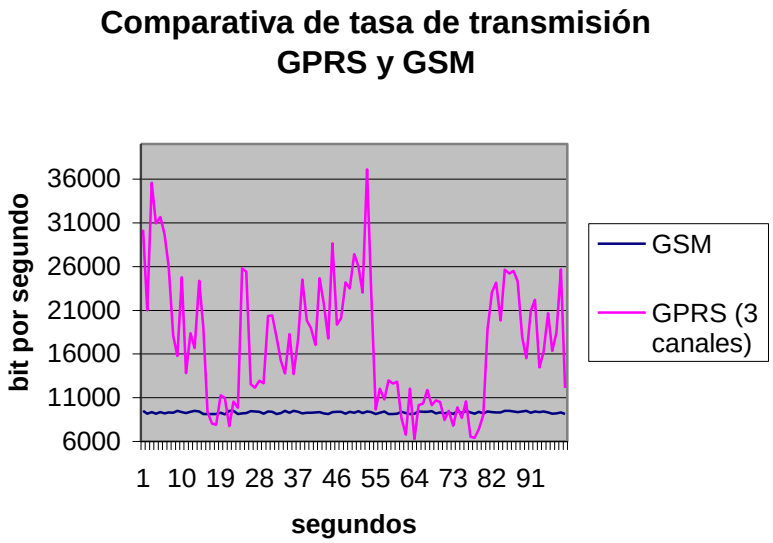

Figura 0.25: Comparativa de tasa de transmisión en GPRS y GSM

El problema con la compartición de los canales es que en caso de que haya muchos usuarios transmitiendo simultáneamente en un momento dado se produce un efecto "desvanecimiento" que provoca una caída en la velocidad de la transmisión. En contrapartida, el hecho de compartir la red con otros usuarios permite que se puedan llegar a mayores tasas de transmisión puesto que se optimiza el uso de la red [31-34].

En general es mejor el uso de la conmutación de paquetes por las mayores velocidades que ofrece y por la posibilidad de facturar por tráfico. Es especialmente recomendable para aplicaciones en las que el tráfico sea discontinuo y no haya consideraciones de tiempo real. En cambio cuando nos enfrentamos a aplicaciones con tráfico continuo y en tiempo real, como *streaming*, las conclusiones no son las mismas. Si con el ancho de banda que nos ofrece la conmutación de circuitos nos vasta, ésta sería la mejor opción. Como esto no suele ser así normalmente se utiliza la conmutación de paquetes, que aunque suele provocar caídas en el flujo de datos, en general da una considerablemente mayor velocidad de transmisión.

2.2.4 Consideraciones sobre seguridad

Como ya hemos visto la seguridad en las comunicaciones celulares digitales es realmente alta. Pero no debemos dar por solucionado el asunto por este motivo. Esta seguridad adicional debemos tenerla

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

en cuenta, pero tenemos que ser conscientes que sólo funciona en un tramo de las comunicaciones inalámbricas.

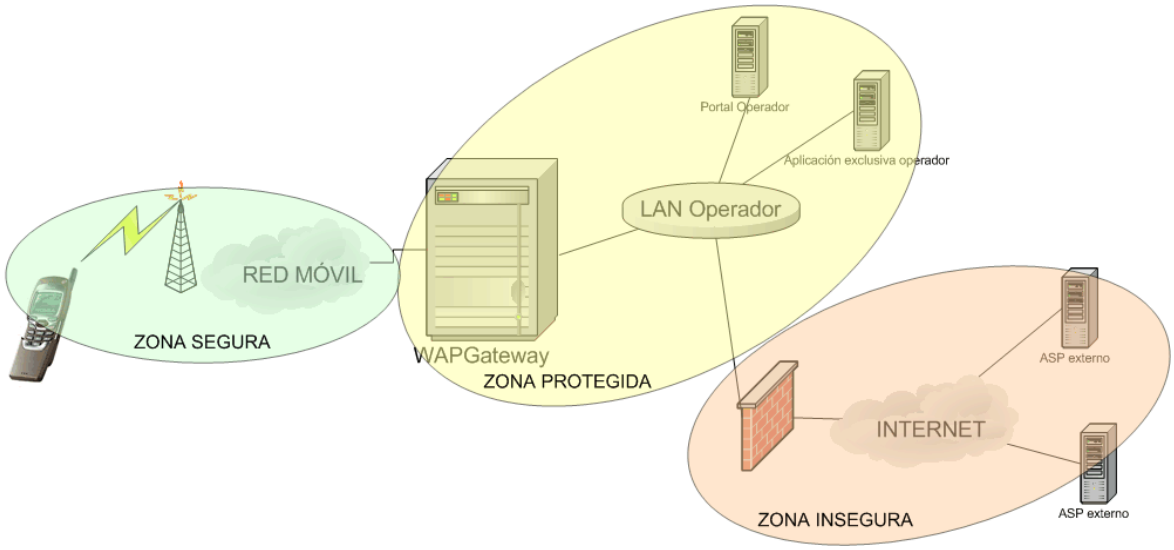

Figura 0.26: Zonas de seguridad en las aplicaciones móviles

Cuando diseñemos una aplicación móvil con acceso a servidores debemos ser conscientes por el número de redes que pasa. Debemos plantearnos qué grado de seguridad queremos. Como se puede ver en el gráfico superior, en el caso de aplicaciones de mensajería o comunicaciones con todas las máquinas servidoras residentes en redes protegidas del operador móvil, la seguridad se puede considerar alta pues en ningún tramo está expuesta a Internet ni a posibles pinchazos (aunque la red del operador no resulta 100% segura).

En el momento que las comunicaciones salgan fuera de la red del operador la cosa cambia y si queremos más seguridad tendremos que buscarla. Una solución más o menos sencilla es establecer un servidor intermedio que establezca comunicaciones seguras con el servidor externo [35-38].

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

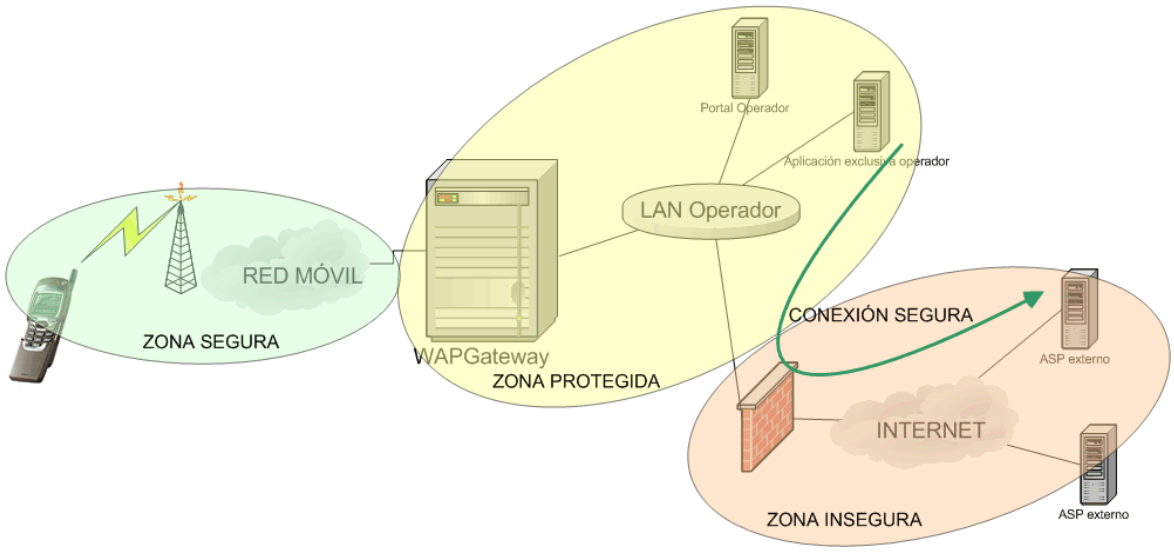

Figura 0.27: Soluciones seguras para las aplicaciones móviles

Aún así esta seguridad puede resultar demasiado limitada para determinadas operaciones de comercio electrónico. En ese tipo de aplicaciones lo óptimo y necesario es realizar una conexión cifrada punto a punto. Y es aquí donde ha estado uno de los problemas más graves en las aplicaciones móviles. Primero porque WAP 1.0 ponía como condición necesaria el uso de un *WAP Gateway* que limitaba la seguridad punto a punto. Segundo porque J2ME MIDP 1.0 tampoco incluía conexiones cifradas.

Una vez detectado este problema se han puesto soluciones en las nuevas versiones de los estándares. El primero que incorporó las conexiones cifradas punto a punto fue el estándar japonés *i-appli*. Posteriormente, J2ME MIDP 2.0 también lo ha incorporado así como WAP 2.0, que evitando el uso del WAP *Gateway* permite las conexiones seguras punto a punto [39-42].

2.2.5 Consideraciones de optimización para los terminales

La gran variedad de modelos de terminal móvil, así como su constante evolución y su gran heterogeneidad provocan que sea realmente difícil realizar una aplicación que esté pensada para un amplio número de modelos y simultáneamente esté lo más optimizada posible para cada uno de ellos.

En el caso de querer realizar una aplicación que llegue a un gran número de modelos y por lo tanto de usuarios, lo más normal es implementarla para funcionar en el caso peor, es decir para que funcione bien en los terminales con capacidades más restrictivas, por lo que la optimización es imposible.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

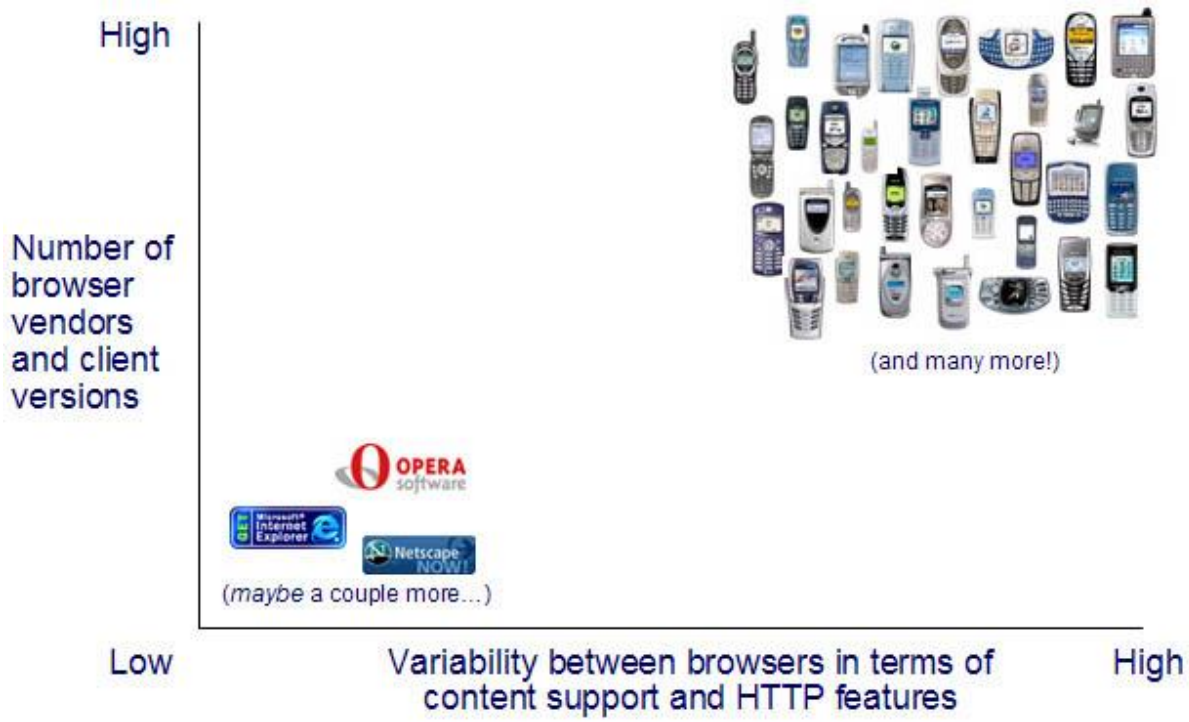

Figura 0.28: Comparación entre la variedad de terminales en el mundo fijo y el móvil

Fuente: OMA

Cuando hablamos de capacidades de terminales, hablamos de cuestiones como la potencia del procesador, la memoria disponible, la implementación de la tecnología utilizada, y sobre todo las capacidades gráficas de la pantalla del terminal. Este último tipo de capacidades son seguramente la más importante a la hora de desarrollar toda clase de aplicaciones, tanto de mensajería multimedia, como páginas WAP o aplicaciones J2ME.

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

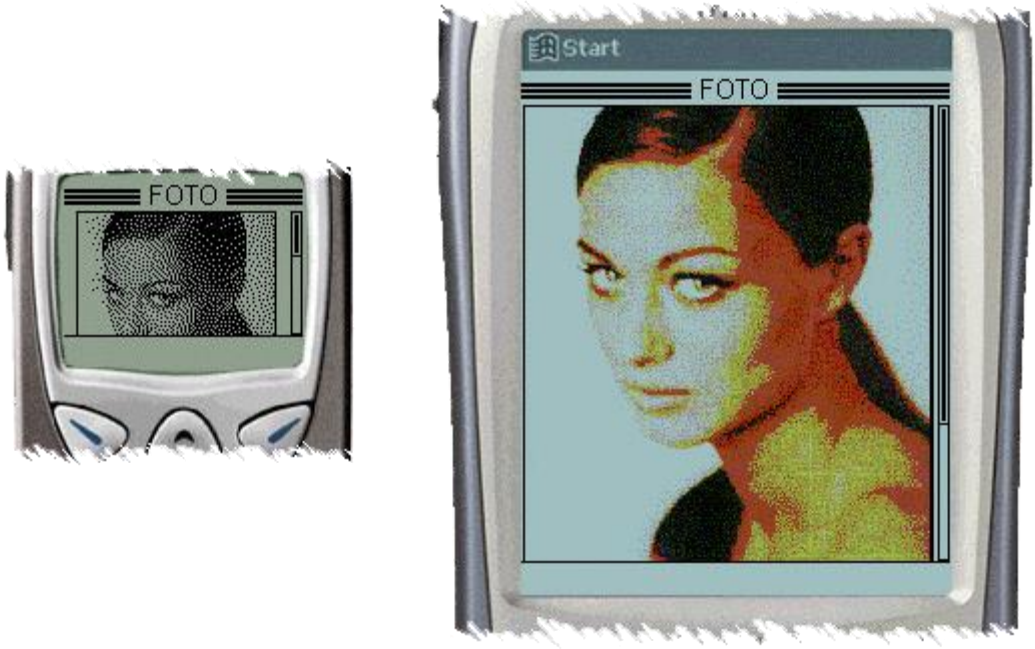

Figura 0.29: Diferencias en la visualización de imágenes en distintos terminales

Las posibles soluciones para este tipo de problemas son dos desde el punto de vista del proveedor de la aplicación:

- Establecer familias de terminales y realizar diferentes versiones de la aplicación para cada una de ellas manteniendo un núcleo común de código.
- Adaptar la aplicación en tiempo real según las características del terminal. Para esto se puede optar por analizar las características mediante el sistema UAProf que analizaremos después y que permite saber las propiedades del terminal.

Por otra parte los fabricantes de terminales y equipos, a través del consorcio OMA ya han recogido estos inconvenientes en diversos estudios y tecnologías. La primera solución optada fue el estándar UAProf, que permitía conocer todas las características del terminal en tiempo real. Evidentemente, esto facilitaba mucho la adaptación de los contenidos.

Aún así esa solución ha distado de ser considerada satisfactoria por varias razones:

- Gran complejidad de muchas adaptaciones, sobre todo de contenidos multimedia
- Son necesarias tecnologías y conocimientos muy avanzados que no siempre tienen los proveedores de contenidos
- Alto coste de creación de contenidos móviles
- Imposibilidad de conseguir adaptaciones correctas por la gran variedad de terminales

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

La línea de acción que está siendo más estudiada actualmente es intentar homogeneizar lo más posible los terminales y el software que incorporan para facilitar el desarrollo de aplicaciones presentando de forma parecida las mismas páginas.

3. MODELOS DE NEGOCIO EN LA TRANSMISIÓN DE INFORMACIÓN EN ENTORNOS MÓVILES

La transmisión de datos en entornos móviles es considerada por muchos la siguiente revolución tanto en Internet como en el mundo móvil. Pero todas las aplicaciones desarrolladas con este fin, se encontrarán y se encuentran con modelos de negocios con particularidades propias y en algunos casos con grandes diferencias con el mundo de Internet [43].

La principal diferencia es la presencia omnipresente y dominante del operador móvil. Es el agente con mayor fuerza en la cadena de valor y el que mayor interés tiene en concentrar todas las aplicaciones y todos los contenidos. Su principal ventaja estratégica reside en su total control sobre la red de acceso y las facturas de los usuarios así como la falta de competidores, otros operadores móviles del mismo país cuyo número suele ser extremadamente reducido. Como veremos más adelante, el modelo más exitoso de transmisión de datos con el móvil corresponde al sistema i-mode, desarrollado y controlado por el operador que más dominio ha ejercido a sus proveedores y a sus clientes, el japonés NTT-DoCoMo.

Además el mundo móvil tiene otras características que modifican los esquemas estratégicos como son la posibilidad de vender terminales preconfigurados con las conexiones del operador de turno, o la habitualidad con que los usuarios aceptan que los operadores sean los que vendan terminales y provean aplicaciones para ellos.

Con estas fortalezas estratégicas los operadores móviles, cuyo negocio original era la provisión de la red, empiezan a posicionarse en otros puntos de la cadena de valor como los portales móviles, la provisión de contenidos o la fabricación y venta de terminales.

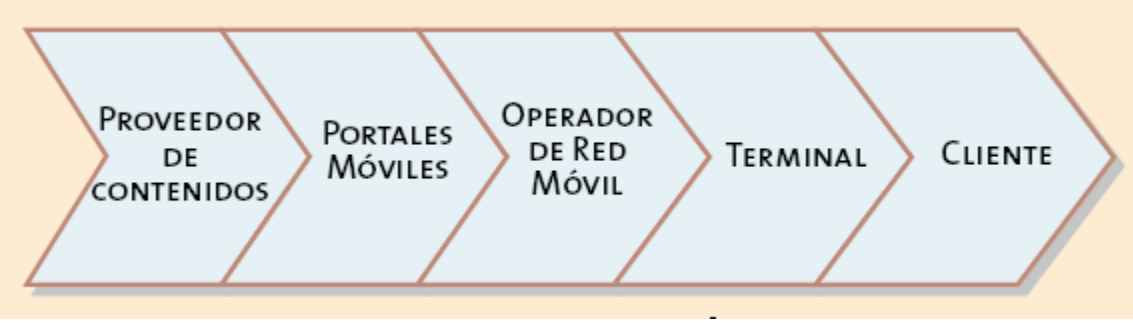

Figura 0.30: Cadena de valor en la transmisión móvil de información

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

En este apartado vamos a ver primero los diferentes modelos de negocio posibles entre los proveedores de contenidos y los operadores para la provisión de aplicaciones y contenidos. También analizaremos los modelos que se han desarrollado para el acceso a estos contenidos, haciendo especialmente hincapié en las correspondientes comparaciones entre el éxito de i-mode, y el fracaso de WAP.

3.1 Modelos de negocio en la provisión de contenidos

En la provisión de contenidos en la industria de la telefonía móvil podemos encontrar los siguientes roles:

- Cliente (usuario de telefonía móvil con cierta avidez de contenidos)
- Operador (proveedor de la conexión y de los elementos de red necesarios para la provisión de contenidos, como por ejemplo el servidor de descargas o el centro de mensajes multimedia)
- Proveedor del servicio (agente que se encarga de proporcionar el portal desde el que se a realizar el descubrimiento de los contenidos por parte del usuario y que prepara la descarga)
- Proveedor de contenidos (agente que se encarga de proporcionar los contenidos para su posterior uso)

Según las empresas cumplan los tres últimos roles podremos establecer los modelos de negocio o escenarios posibles en la provisión de contenidos:

- En el primer modelo de negocio, el proveedor del contenidos se convierte también en proveedor del servicio (o viceversa), y el operador se reservaría el elemento de red que facilitaría la descarga. De esa forma ésta se produciría de forma segura y el servicio tendría una facturación eficiente y sencilla. Evidentemente se tendrían que llevar a acuerdos con los operadores para compartir los ingresos correspondientes a los contenidos. En este modelo se podría dar el caso que los proveedores de servicios alquilasen al operador el servicio de descargas o que se fuese a un modelo de compartición de beneficios.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241

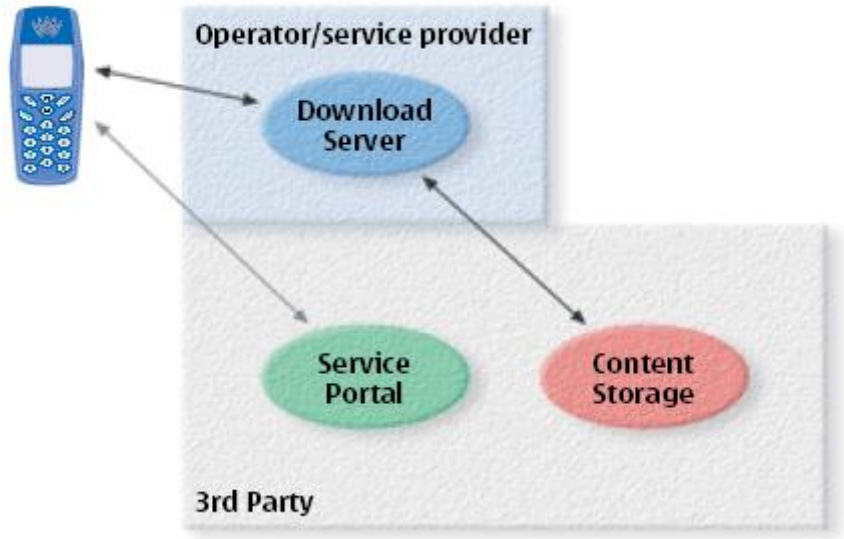

Figura 0.31: Modelo de negocio con el operador como *carrier* Fuente: Nokia

- En un posible segundo modelo de negocio, los operadores actuarían como agregadores de contenidos. Además del mecanismo de descarga y facturación aportarían al sistema el portal del servicio que, de hecho, podría ser común a varios proveedores de contenidos. De esta forma el operador podría conseguir la exclusividad del servicio frente a usuarios de otros operadores. Además el modelo sería más eficiente económicamente puesto que el portal sólo tendría que ser desarrollado y mantenido una sola vez y lo podrían utilizar los proveedores de contenidos que autorizase el operador. Evidentemente se tendrían que llevar a acuerdos con los operadores para compartir los ingresos correspondientes a los contenidos.

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

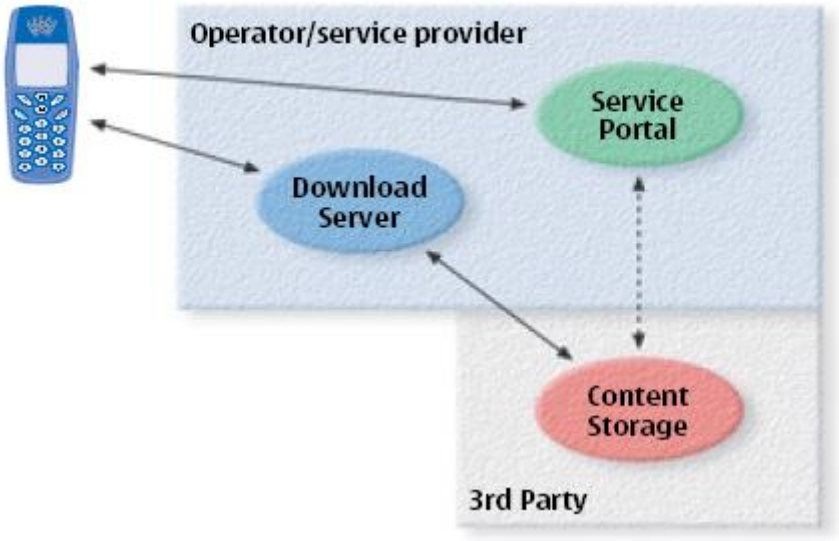

Figura 0.32: Modelo de negocio con el operador como agregador de contenidos Fuente: Nokia

- Finalmente en el tercer modelo, el operador crea o compra los contenidos y provee totalmente el servicio. En este escenario es posible que la compartición de beneficios no fuese la opción a elegir sino que el operador pagaría por los contenidos de forma independiente al éxito de estos posteriormente.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

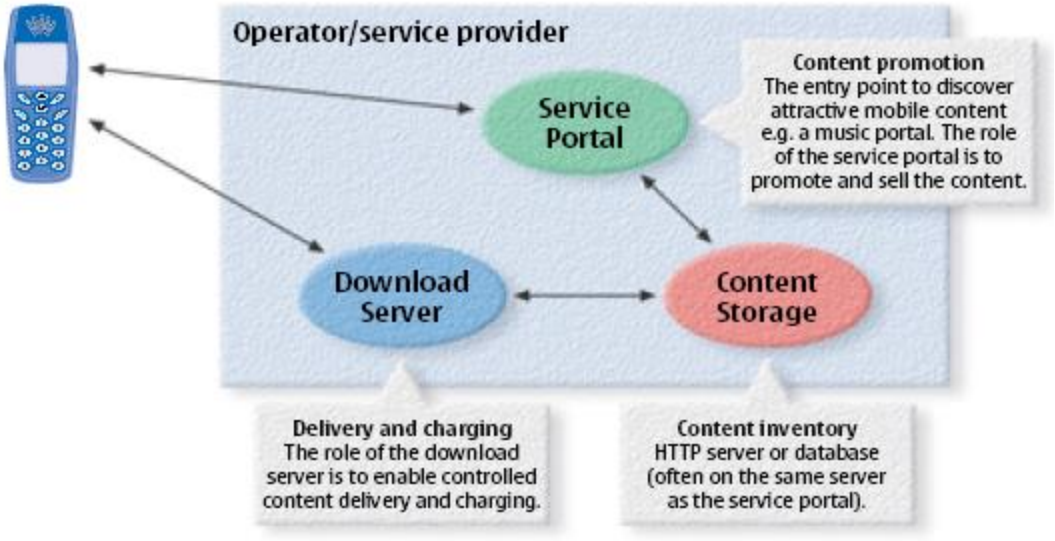

Figura 0.33: Modelo de negocio con el operador como proveedor de contenidos Fuente: Nokia

3.2 Modelos de negocio en el acceso a los contenidos

En este apartado vamos a tratar de analizar por qué el modelo tecnológico y empresarial de NTT-DoCoMo (propietaria de i-mode) ha resultado ser muy superior (por lo menos en resultados) que el desarrollado por la operadoras que han apostado por WAP. Primero analizaremos a fondo la estrategia y el éxito de i-mode para posteriormente compararlo con WAP y finalmente, para sacar las conclusiones pertinentes.

3.2.1 I-mode

No se puede simplificar el concepto de i-mode y por lo tanto no es fácil definirlo en pocas palabras. Va más a allá de un conjunto de servicios o una especificación técnica, se podría decir que es un modelo de explotación integral del acceso a contenidos de Internet mediante el móvil. También se podría decir que es una forma o filosofía de llevar la tecnología hasta los usuarios.

Nace en 1999 de la mano de la operadora móvil japonesa NTT-DoCoMo. De forma resumida y como veremos incompleta, se puede decir que nace un servicio similar a WAP en el que el lenguaje de programación es c-HTML, un lenguaje de marcado reducido basado en HTML y en el que las comunicaciones se basan en estándares propietarios que permitían en esa época velocidades de hasta 9,6 kbps mediante conmutación de paquetes.

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

A partir de esa fecha inicial, el crecimiento ha sido espectacular. No sólo en clientes subscritos, sino también en el aumento de los servicios ofrecidos y en la mejora de las tecnologías utilizadas. Aparecen los teléfonos a color, con soporte para i-appli (un J2ME mejorado), cámaras de fotos, nuevos servicios como i-motion (videos) o iarea (localización). También mejoran substancialmente las comunicaciones llegando a 28,8 kbps e incluso a 384 kbps con FOMA, la apuesta 3G de NTT-DoCoMo.

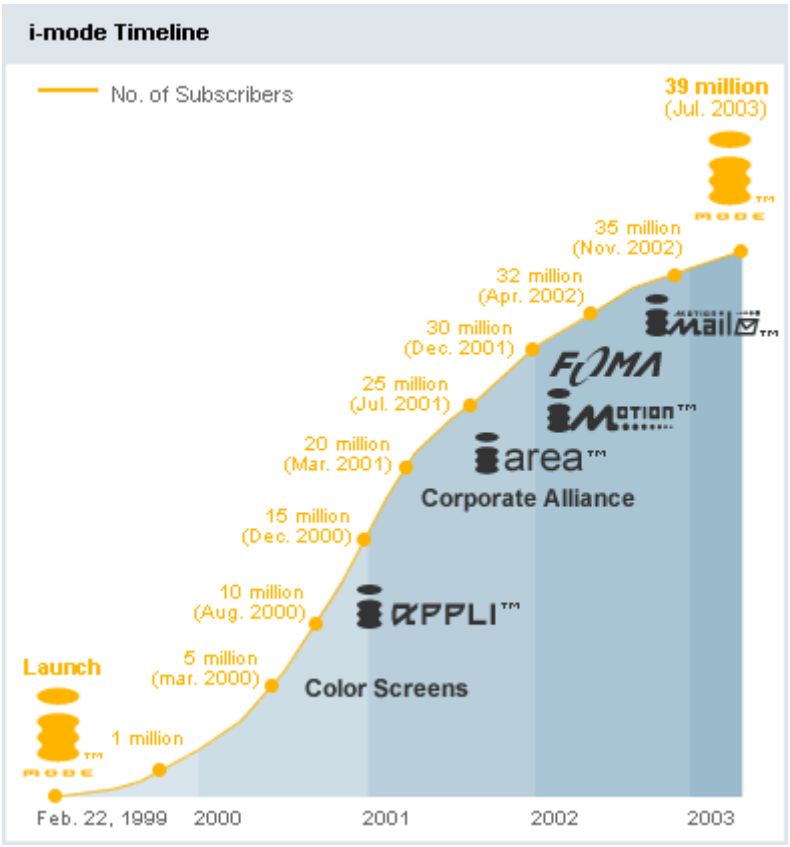

Figura 0.34: Evolución de los clientes y servicios de i-mode Fuente: NTT-DoCoMo

No vamos a detallar en profundidad las tecnologías utilizadas ni los servicios publicados. Nuestro interés en este apartado es ver cómo un modelo de negocio puede significar un éxito rotundo. Para ello vamos a analizar como funciona el modelo de negocio de i-mode para luego estudiar las razones del éxito abrumador de i-mode en Japón [44].

I-mode opera bajo un modelo de negocio en el que el usuario, después de darse de alta con una cuota de 300 ¥/mes (2,6 €/mes), puede acceder a cualquiera de los más de 3.000 sitios registrados oficiales de más de 900 proveedores de contenidos (el 30% ofrecen contenidos *premium* mediante suscripción), listados en orden de popularidad bajo diferentes menús y categorías. El usuario además abona

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

0,3 ¥/130 bytes (0,026 €/ 130 bytes) por el volumen de datos trasmitidos o recibidos (por ejemplo, un parte del tiempo puede costar en torno a un tercio de euro).

Al ser un servicio proporcionado directamente por el operador móvil, los subscriptores no necesitan facilitar un número de tarjeta de crédito para efectuar sus compras o abonarse a los contenidos *premium* (en los sitios oficiales). El coste de los servicios *premium* de los sitios oficiales (el coste suele estar entre 100-500 ¥/mes o 1,15-4,3 €/mes) y de los productos adquiridos aparece directamente en la factura del abonado y DoCoMo revierte este dinero al proveedor de contenidos, no sin antes retener para sí un 9% del total en concepto de comisión.

En el caso de los sitios no oficiales (más de 60.000), los contenidos o los productos que requieran de pago por parte del usuario deberán ser abonados mediante tarjeta de crédito o cualquier otro mecanismo decidido y proporcionado por el sitio no oficial. En este caso DoCoMo sólo recibe el dinero correspondiente al tráfico generado. Además estos sitios no tienen porque cumplir las estrictas normas del operador y pueden ofertar contenidos no autorizados (contenidos para adultos, ejemplo) y además pueden establecer tarifas de suscripción mayores de 500 ¥. De todas formas, más de dos tercios de estas páginas son gratuitas.

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

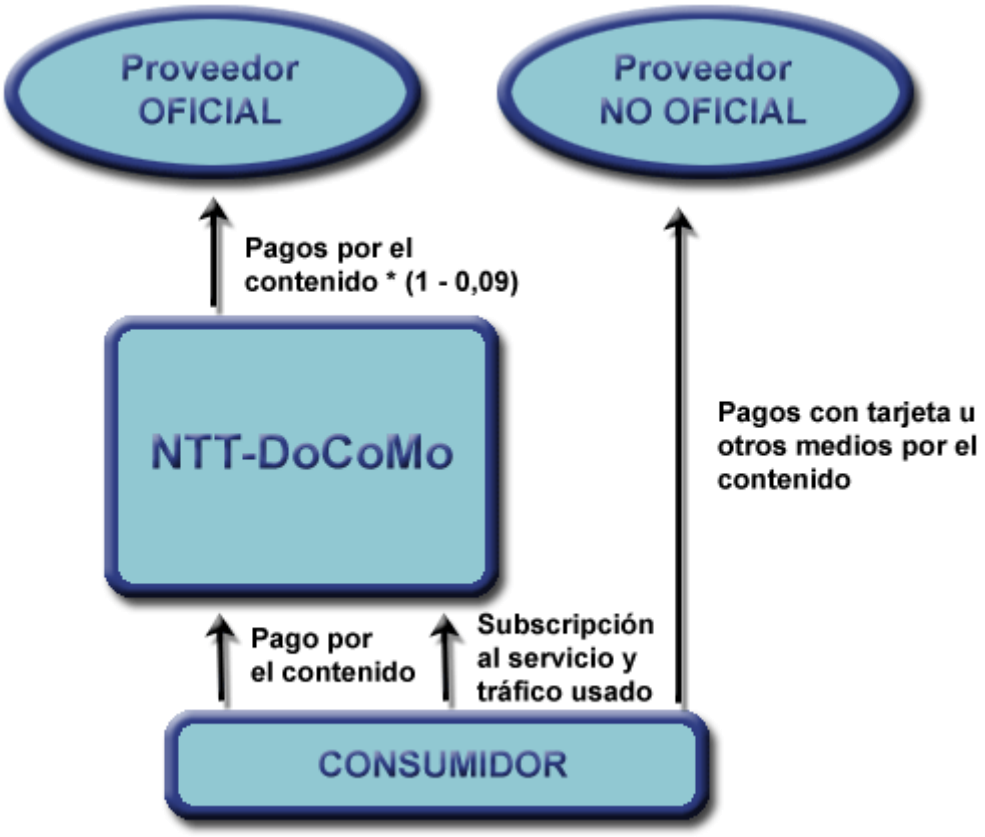

Figura 0.35: Diagrama del flujo del dinero en los servicios i-mode

Las razones del éxito de NTT-DoCoMo son varias, aunque la mayoría se pueden resumir en la capacidad de la compañía para coordinar todas las etapas de la cadena de valor y responder a las necesidades del cliente final. Podemos dividir las razones del éxito de i-mode entre las achacables al operador y las causadas por las especiales características del mercado. Empecemos por las primeras.

- NTT-DoCoMo siempre ha dado una alta prioridad a la simplicidad y usabilidad de los servicios sobre la innovación. Las decisiones técnicas siempre estaban supeditadas a la necesidad de desarrollar aplicaciones que fuesen útiles a los usuarios desde el primer momento.
- NTT-DoCoMo ha mantenido un férreo control sobre los terminales del servicio. Siempre ha procurado que fuesen los terminales los que se adaptasen a los contenidos, y no al contrario. La compañía ha trabajado de forma muy cercana con los proveedores de los terminales, especificando todas las características técnicas de los dispositivos. De esta forma, los terminales se comercializan bajo la marca de NTT-DoCoMo, aunque la primera letra del modelo indique el fabricante (N503i es de Nec, F503i es de Fujitsu). Finalmente, cuando se lanzó el servicio ya se contaba con un buen número de terminales en manos de los usuarios pues se había subvencionado fuertemente los teléfonos.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

- La operadora japonesa ha creado un departamento con más de 60 personas para ya ayudar a los proveedores de contenidos y educar al usuario final sobre las oportunidades de negocio de la plataforma i-mode.
- Además, también utiliza a un riguroso proceso de selección para dar la calificación de "oficial" a una página i-mode. Aún así no desanima a los 1.000-6.000 candidaturas mensuales que recibe la compañía y provoca que haya actualmente una lista de espera de más de 6 meses.
- Además del control sobre los fabricantes de terminales y los proveedores de contenidos, NTT-DoCoMo también mantiene unas fuertes relaciones con los canales de distribución. Por un lado dispone de una importante red de tiendas propias que proporcionan un canal directo óptimo. Pero además también se relaciona intensamente con los canales indirectos, todas las tiendas de electrónica, extremadamente populares en Japón. Para motivar la suscripción al servicio, NTT paga a estas tiendas 17 € por cada usuario que se registra al servicio cuando compra un terminal. Esta comisión es superior a lo que se suele dar en Europa o América y provoca que los comercios estén realmente motivado en la venta del servicio con lo que se estima que entre el 20% y el 40% de los usuarios de i-mode sólo usa el terminal para servicios de voz.
- Una de las facetas que más ha apoyado NTT-DoCoMo es el de las alianzas estratégicas con empresas líderes. Claros ejemplos son las alianzas con AOL, Sony o Disney.
- La elección del mercado objetivo y la forma de promocionar el servicio han sido perfectas. Por un lado optaron por orientarse a un público joven, menor de 30 años, con menor resistencia al cambio y mayor costumbre con la tecnología (mediante i-mode un usuario medio envía 8 e-mails y visita 11 páginas al día). Además nunca promocionaron el servicio como Internet en el mundo móvil, cualquier referencia a "tecnología", "web" o "navegador" fue cuidadosamente evitada.

Evidentemente, Japón no es un mercado normal o comparable con el resto, especialmente en cuando hablamos de electrónica o entretenimiento. Estas especiales características han influido considerablemente en el éxito de i-mode.

- La penetración de los móviles era muy alta.
- Características de la cultura japonesa, largos desplazamientos en tren, perfil entusiasta ante las innovaciones tecnológicas, carácter cerrado de los japoneses que evitan contactos directos, etc..

3.2.2 WAP

WAP es sencillamente un conjunto de estándares. Incluyen una pila de protocolos para realizar las comunicaciones móviles y un entorno de ejecución de aplicaciones en el terminal. En ningún caso se entró en la forma de incorporar esta tecnología desde el punto de vista de negocio o marketing. Estos apartados siempre se reservaron para los operadores de cada país.

Cada operador tuvo la libertad (y la sigue teniendo) de lanzar las aplicaciones que quiera, al precio que quiera y en el momento deseado. Actualmente WAP sólo define los medios técnicos necesarios para hacerlo, un poco al estilo de Internet. Como podemos imaginar, la situación es muy diferente a la de i-mode. Los operadores han tenido unas relaciones mucho menores con los proveedores de

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

contenidos y los fabricantes de terminales. Además el modelo de negocio está totalmente abierto. Aún así, la mayoría de operadores han optado por cobrar la conexión según el tiempo o el tráfico utilizados y poniendo un suplemento (*premium*) a los contenidos más interesantes o valiosos. Las tarifas planas han brillado por su ausencia (aunque algún intento ha habido) por lo que los usuarios siempre han tenido que pagar proporcionalmente al uso de las aplicaciones.

Desde los primeros lanzamientos comerciales del año 2000, ningún operador ha conseguido encontrar la clave del éxito de las aplicaciones sobre WAP. Partiendo de las expectativas generadas en esa época, seguramente desmesuradas y poco realistas, para el año siguiente ya se empezaba a hablar de un rotundo fracaso de uso. Las razones de las dificultades que encontraron las aplicaciones WAP en su momento son varias:

- Al contrario que en Japón, donde había un gran número de terminales con soporte i-mode en su lanzamiento, los operadores europeos lanzaron servicios WAP antes que hubiese un mínimo número de terminales en manos de los usuarios. Además, durante bastante tiempo no hubo terminales con atractivo diseño, capacidades técnicas avanzadas y a un precio razonable.
- La orientación que se le dio en la mayoría de países fue Internet en el móvil. No sólo se buscó adaptar las aplicaciones de Internet al móvil sin aportar más valor añadido, sino que además se promocionó y publicitó de esa forma, llevando a la frustración y la decepción a los primeros usuarios.
- Falta de contenidos atractivos y servicios demandados por usuarios. Mientras NTT-DoCoMo se preocupaba de incluir contenidos interesantes y con un alta calidad, los operadores europeos han pecado de dejadez en este terreno.
- En un comienzo las comunicaciones se realizaron sobre GSM y conmutación de circuitos. Esto provocaba que los tiempos de acceso fuesen realmente altos y que se cobrase por el tiempo de uso con lo que los usuarios nunca se atrevieron a utilizar masivamente las aplicaciones.

Lo cierto es, que pasados varios años desde su lanzamiento WAP empieza a ver la luz. Por un lado las mejoras de las tecnologías, tanto de las de transmisión, como las de los terminales, han permitido poder realizar aplicaciones más interactivas, rápidas y espectaculares llegando además a un mayor número de usuarios. Además, los operadores han replanteado estrategias y actualizado expectativas analizando y actuando con mayor tranquilidad y realismo.

De alguna forma u otra, las aplicaciones de Internet tendrán que ser usadas a través del móvil. El número de usuarios de telefonía móvil (altísimo) y la mayor familiaridad de la ciudadanía con las aplicaciones de Internet han de provocar que tarde o más bien temprano, el uso de móvil para conectarse a Internet se extienda. Y WAP parece bastante bien posicionado para ser la tecnología utilizada en un buen número de países.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

DESCARGA DE CONTENIDOS

La descarga de imágenes, sonidos, películas, programas o documentos es uno de los mayores usos de Internet actualmente. El incremento del ancho de banda y el uso de PC como terminales de conexión han permitido que la cantidad y la variedad de las descargas de archivos de la red sean muy elevadas.

En el mundo móvil la situación es un poco diferente. Por un lado las descargas son menores en cantidad y calidad. Lo que más abundan son melodías e imágenes de baja calidad y tamaño para personalizar el terminal, pero esto se compensa con los resultados económicos de estas descargas. Estos ingresos han sido realmente espectaculares, tanto para los operadores como para los proveedores de los contenidos. De hecho, tal y como podemos ver en el siguiente gráfico, las descargas realizadas a través de teléfonos móviles suponen un negocio más de dos veces mayor que las correspondientes a los terminales PC.

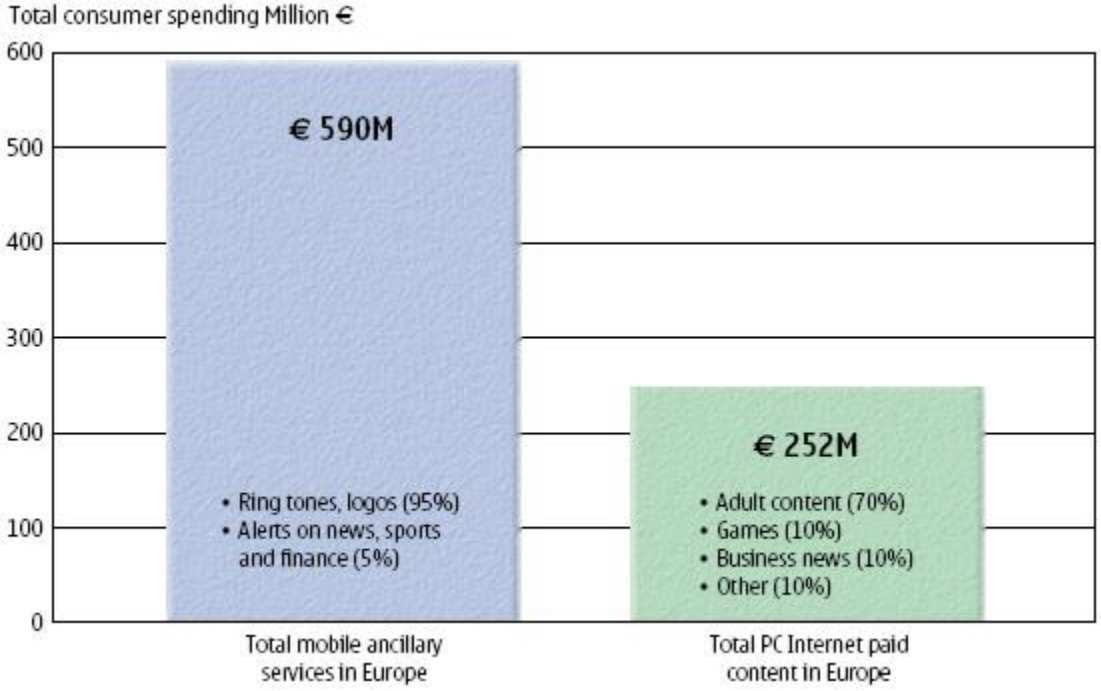

Figura 0.36: Ingresos en las descargas de contenidos Fuente: Jupiter

Las razones de este éxito han sido varias:

- Por un lado las facilidades tecnológicas que ofrece la telefonía móvil para este tipo de aplicaciones. Al usuario le resulta muy sencillo mandar un mensaje o realizar una llamada para recibir posteriormente un mensaje con el contenido deseado.
- Las facilidades que aporta el móvil a la hora de realizar micro-pagos o pagos de pequeña cuantía. Gracias a la posibilidad de incluir el cobro de los contenidos en la factura del usuario

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

ha resultado muy sencillo cobrar al usuario todas las descargas realizadas. Además ha permitido crear una cultura que no existe en Internet, que permite acostumbrar al usuario a pagar por los contenidos descargados. Esto va a tener gran importancia en las posibilidades de éxito de los futuros planes de negocio para los servicios móviles.

- El terminal móvil es un objeto personal que además se suele llevar permanentemente. Esto provoca que los usuarios quieran personalizar su terminal así como personalizan el resto de su imagen exterior. De esta forma es fácil comprender el uso de las descargas móvil con objeto de dotar al terminal de las imágenes y tonos más modernos o curiosos.
- Dificultad para compartir los contenidos entre los usuarios. En los primeros años de las descargas de tonos y logos era razonablemente complejo mandar estos contenidos con otros usuarios, esto provocaba que fuese más fácil y barato comprar por un precio bajo los archivos.

Estos positivos resultados de los últimos años, junto con la evolución constante de las capacidades de los terminales permiten pronosticar un importante éxito empresarial en el futuro. Además uno de los principales problemas que en el pasado han existido ha sido la gran variedad de tecnologías de descarga disponibles en el mercado. Nokia creó el *Smart Messaging* sobre los SMS para mandar contenidos, además también empezó a especificar una tecnología para la descarga segura sobre WAP. Otros fabricantes hicieron lo propio, por ejemplo, Openwave con su tecnología *DownloadFun* fue pionera en la descarga de contenidos mediante conexiones WAP.

Afortunadamente, tanto los fabricantes de teléfonos, como los desarrolladores de aplicaciones móviles y navegadores para los terminales se han puesto de acuerdo para empujar desde el principio el consorcio OMA (*Open Mobile Alliance*). Este consorcio tiene entre trabajos la especificación de las tecnologías para desarrollar aplicaciones basadas en descargas de contenidos desde un terminal móvil.

A simple vista, las tecnologías involucradas para la descarga de contenidos ya deberían estar superadas. En la WEB se producen todos los días millones de descargas mediante el protocolo HTTP y en el entorno móvil además ya existe la mensajería multimedia. Pero la utilización solitaria de estas tecnologías provoca tres grandes problemas:

- Un usuario puede intentar descargar un contenido que no cabe en su terminal o que no acepta el terminal porque el formato es demasiado avanzado. El usuario no tiene porque tener conocimientos sobre formatos gráficos y en caso de realizarse la descarga posteriormente no se podría instalar o utilizar el contenido. Esto provocaría que o bien se le cobra al usuario un contenido que no podrá usar (con la consiguiente mala experiencia para el cliente) o bien se genera un tráfico que luego no se factura. Cualquiera de estas situaciones no deseable en un entorno en el que se cobra el contenido o el tráfico o ambos.
- Un problema más grave es si el terminal sufre una desconexión a mitad de la descarga o por cualquier otra causa no puede descargar del todo el contenido e instalarlo si procede. En ese caso estaríamos en una situación parecida a la anterior siendo otra vez necesario no cobrar al usuario por un contenido que no podrá utilizar.
- Finalmente también sería deseable que se protegiesen los contenidos de pago para que el usuario no pudiese mandarlos fácilmente. Con lo terminales más avanzados estas posibilidades cada vez son más fáciles. Además, también sería muy interesante poder establecer polí-

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

ticas para el uso de los contenidos. Por ejemplo, sería muy interesante poder especificar cuántas veces se puede utilizar un contenido o durante cuánto tiempo, de esa forma se podrían hacer varias tarifas o realizar descargas de prueba a modo de previsualización.

Para solucionar estos puntos se han desarrollado nuevas tecnologías complementarias a WAP y a MMS. Por un lado se ha diseñado lo que se denomina *OMA Download*, que no es sino un conjunto de estándares para solucionar los dos primeros problemas antes comentados. En el próximo apartado analizaremos sus puntos más importantes.

En el segundo apartado entraremos a estudiar la solución que en el consorcio OMA se ha dado al tercer problema, la protección de los contenidos. La tecnología propuesta para realizar esta labor se denomina DRM (*Digital Rights Management*)

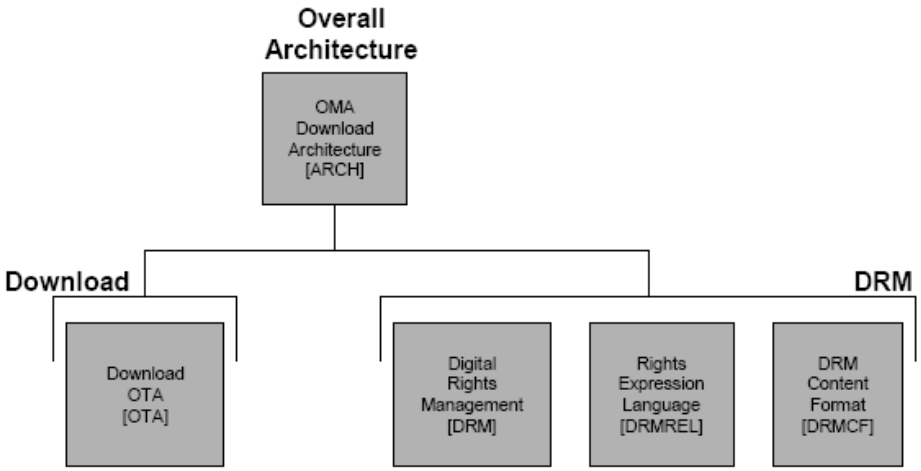

Figura 0.37: Arquitectura global de las descargas Fuente: OMA

3.3 OMA Download

Como hemos visto, la descarga de contenidos se debe realizar mediante tecnologías que permitan realizar transacciones robustas entre el terminal y el servidor de descargas con el fin de proporcionar al usuario la mejor y más sencilla experiencia de cliente posible y al operador o proveedor de servicios la posibilidad de cobrar el contenido de forma simple y con protección de copia.

Para realizar estas funciones la industria de la telefónica móvil, a través del consorcio OMA, ha definido el estándar *OMA Download* para facilitar la descarga de todo tipo de contenidos, como melodías MIDI o salvapantallas en formatos JPEG, GIF animados o PNG entre otros muchos tipos de formatos y contenidos. Las ventajas de esta tecnología son varias:

- Evita la fragmentación de las aplicaciones de descarga que se podría producir si todos los fabricantes de terminales implementasen sus propias tecnologías de descarga. En ese escenario los desarrolladores de aplicaciones de descarga tendrían que primero detectar qué terminal

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

está intentando descargar el contenido y luego enviárselo según la tecnología que correspondiese.
- Basada en la descarga de contenidos a través de HTTP, las tecnologías que involucra y la arquitectura que tiene son realmente sencillas, permitiendo desarrollar servicios de descarga en unos plazos de tiempo realmente cortos.
- Permite realizar extensiones sobre ella, como el caso de DRM, facilitando crear mecanismos de gestión de los derechos de los contenidos.

La primera tecnología de descargas y la más sencilla es la descarga HTTP. En ella el cliente realiza una petición HTTP al servidor indicándole el contenido a descargar y éste contesta mandando el contenido e indicando con el tipo MIME la clase de contenido que es. *OMA Download* se basa en este tipo de descargas para realizar las diferentes fases de la que se compone. Pero además, su arquitectura es muy similar a las descarga de aplicaciones J2ME, es decir a la tecnología *Java MIDLet Download.*

La tecnología *Java MIDLet Download* es más antigua que *OMA Download* y se creó para facilitar las descargas de aplicaciones J2ME. El modelo de uso es exactamente igual al de *OMA Download* con algunos cambios que iremos viendo según vayan apareciendo. Lo que es evidente, es que la descargas especificadas por OMA surgen de inspirarse en las descargas de *MIDLets* que se habían demostrado eficaces y sencillas de implementar.

Vamos a dividir el estudio de *OMA Download* en tres apartados. Primero veremos las diferentes arquitecturas que puede adoptar, después veremos las distintas fases de una descarga y luego veremos el formato del archivo que describe el contenido y sobre el cual gira toda la especificación. Durante todo el análisis de esta tecnología debemos tener en cuenta que para realizar la descarga el usuario estará navegando por páginas en las que se listarán todos los contenidos descargables. Estas páginas las generará el servidor de presentación que accedería al repositorio de contenidos para saber cuáles hay. Cuando el usuario decida descargarse alguno realizará una petición al servidor de descargas el cuál será el que implemente *OMA Download* y el cuál también accederá al repositorio de contenidos para facilitar la descarga.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

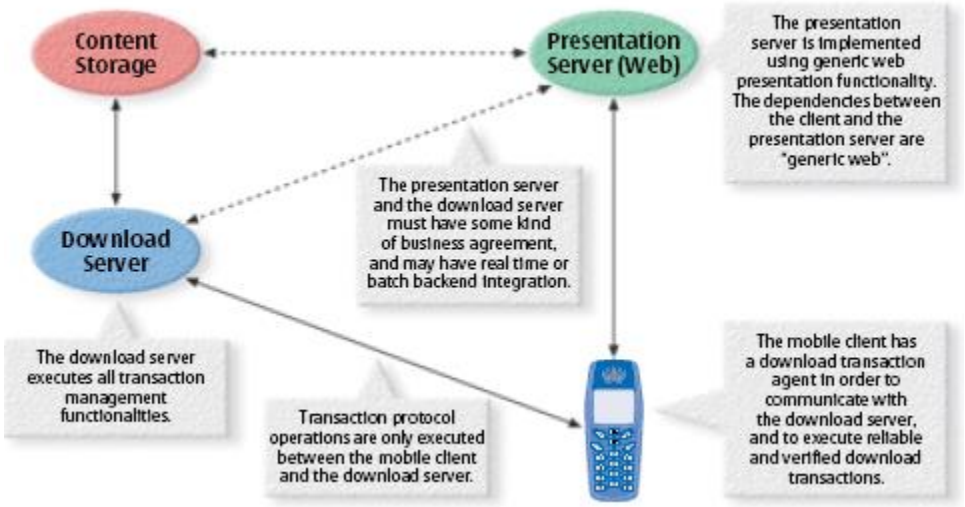

Figura 0.38: Elementos que intervienen en un proceso de descargas Fuente: Nokia

3.3.1 Arquitectura

OMA Download se basa en varias peticiones HTTP para realizar toda la descarga. En esas peticiones se realizan principalmente tres acciones:

- Descarga de un descriptor del contenido. Más adelante veremos cuál es el formato de este descriptor, pero como veremos tendrá información tal como el formato del contenido, su tamaño, etc.
- Descarga del propio contenido.
- Notificación al servidor de descargas de que el contenido ha sido correctamente descargado e instalado.

El último punto es optativo. Además los dos primeros se pueden realizar en dos peticiones HTTP diferentes o en una sola. De esta forma tendremos dos posibles arquitecturas.

- *OMA Download* con descarga separada del contenido y su descriptor

  En esta arquitectura cuando el usuario selecciona el contenido a descargar la petición que se realiza al servidor de descargas es para obtener el descriptor del contenido exclusivamente. En este descriptor se informa de la URL del contenido que se desea descargar. Si el terminal decide posible la descarga y el usuario la confirma se realizará otra petición con el objeto de descargar realmente el contenido.

  Finalmente, se realizará una petición al servidor de descargas indicando el éxito o fracaso de la descarga. Para realizar esta petición también se obtendrá la dirección del descriptor anteriormente obtenido.

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

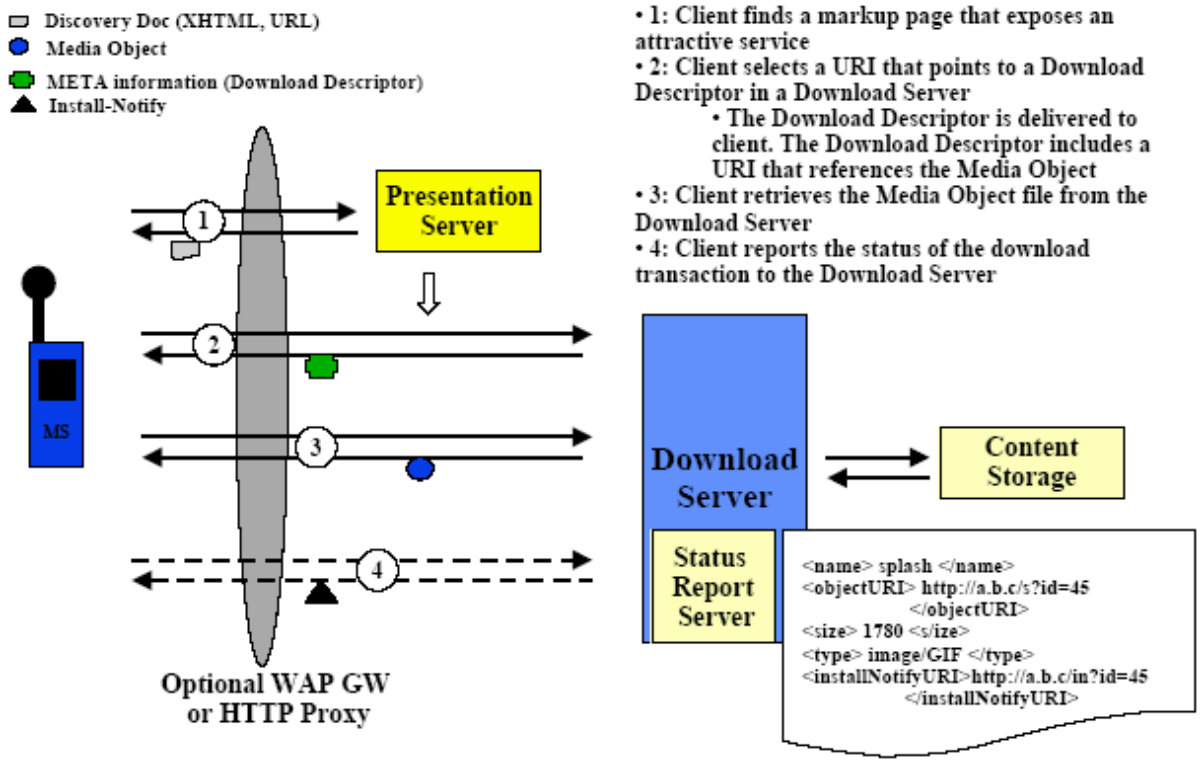

Figura 0.39: Arquitectura de descargas OMA con envío separado de descriptor y contenido Fuente: OMA

- *OMA Download* con descarga conjunta del contenido y su descriptor

En esta arquitectura el envío del contenido y su descriptor se realiza conjuntamente en una sola petición-respuesta HTTP. Este modelo es utiliza considerablemente menos, porque apenas aporta ventajas y supone perder parte de los beneficios esperados, como por ejemplo la comprobación previa del terminal sobre el formato y tamaño del contenido.

Para la realización del envío simultáneo del contenido con su descriptor se debe usar el formato *multipart/related* de HTTP.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

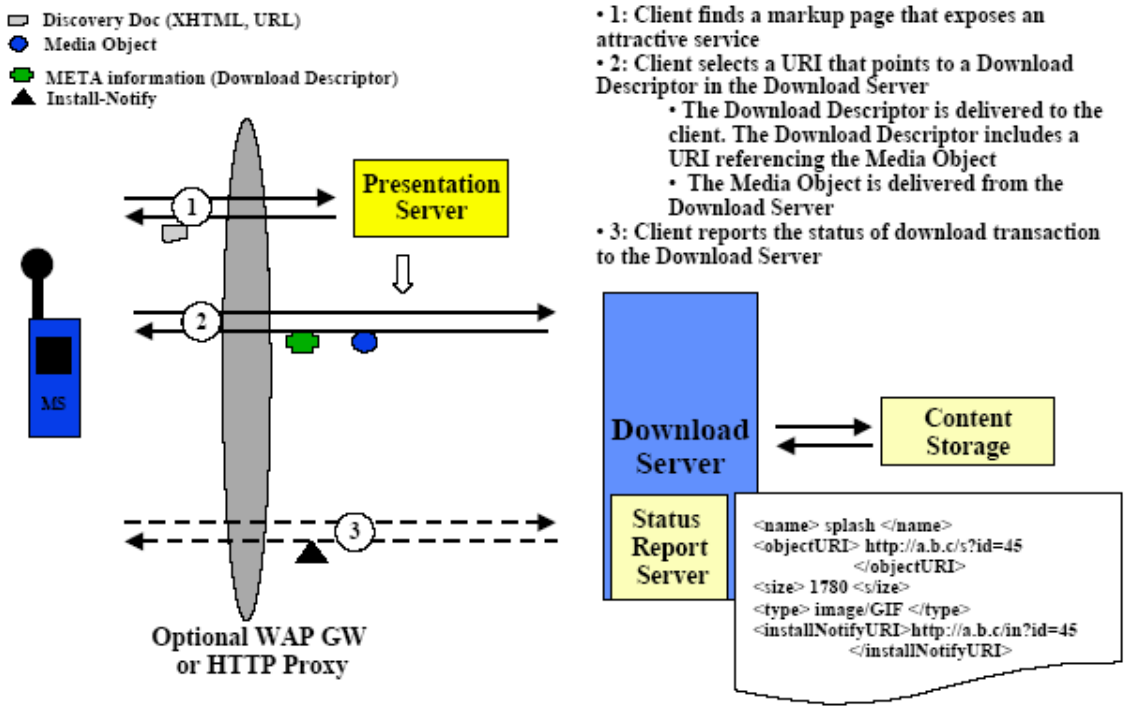

Figura 0.40: Arquitectura de descargas OMA con envío conjunto de contenido y descriptor Fuente: OMA

En ambas arquitecturas el descriptor contiene una URL contra la que al final se puede realizar una petición informando del estado final de la descarga. Aún así esta funcionalidad es optativa y se puede considerar la arquitectura sin necesidad de realizar esta petición. En este caso sacrificaríamos la confirmación de la descarga por una mayor optimización del ancho de banda usado.

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

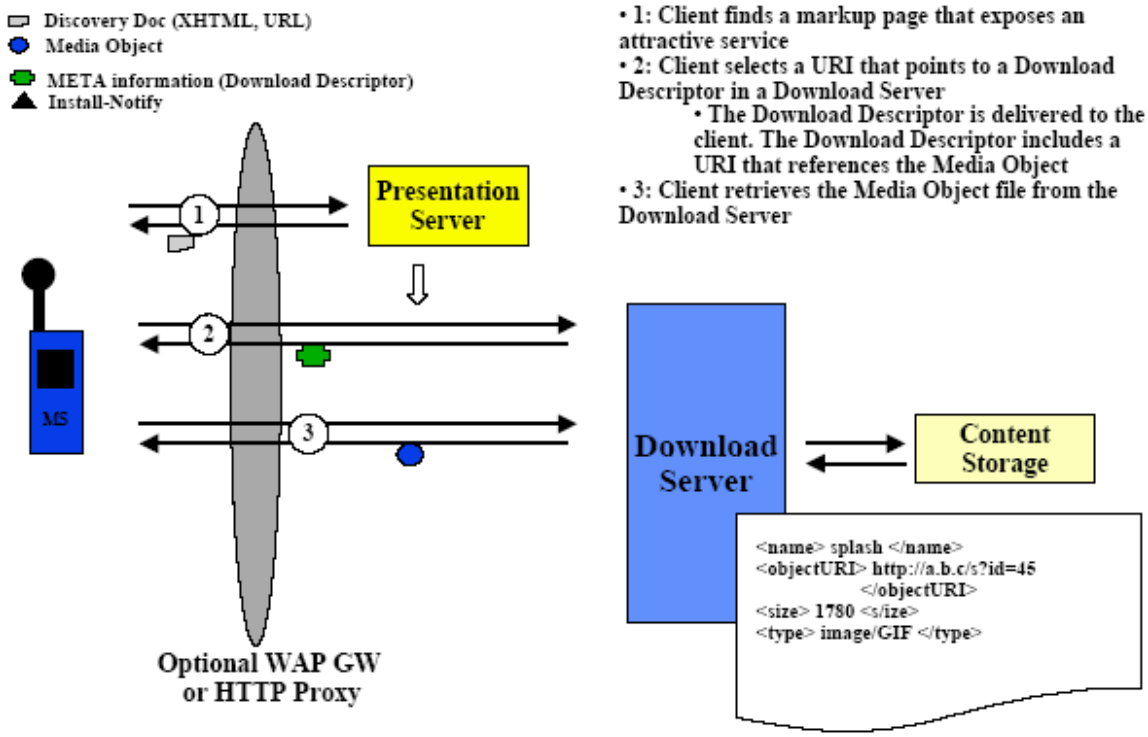

Figura 0.41: Arquitectura sin confirmación de descarga de contenido Fuente: OMA

3.3.2 Fases de la descarga

Durante todo el proceso de la descarga se pueden identificar diferentes fases en las que se van realizando diferentes acciones o peticiones para completar con éxito toda la transacción.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

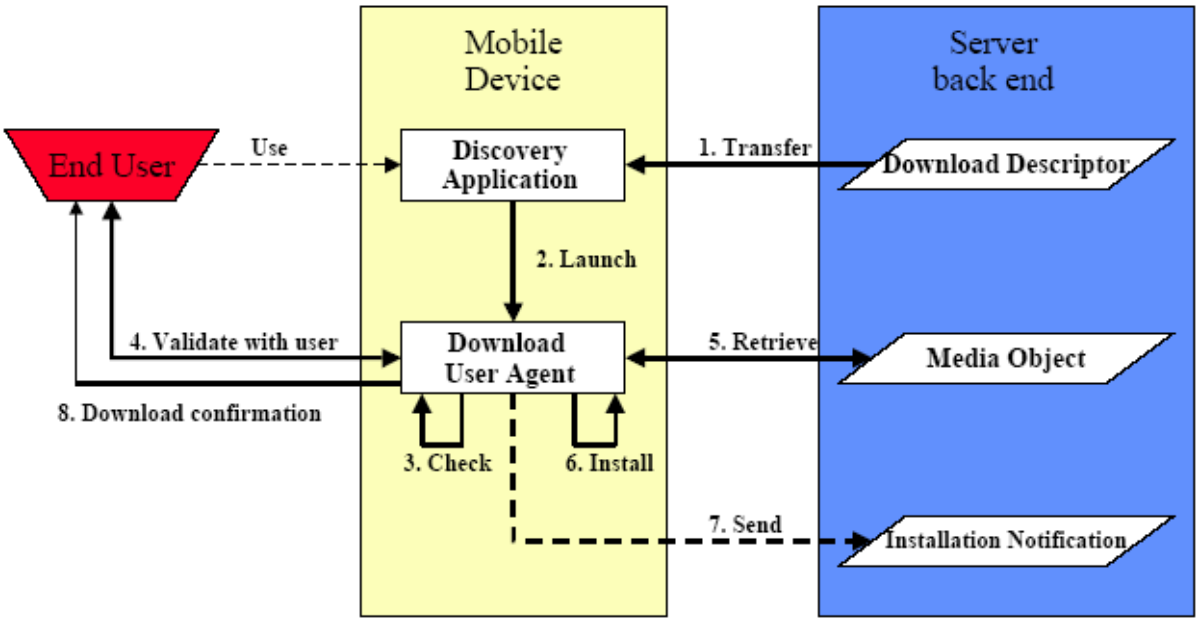

Figura 0.42: Fases de la descarga Fuente: OMA

A partir de la segunda fase, el terminal ya ha obtenido el descriptor del contenido. Con este descriptor se puede saber si hay que notificar el éxito de la descarga o cualquier error que suceda. En las siguientes pases posteriores a la primera cualquier error que suceda podrá ser notificado a la dirección de notificación mediante del envío de una petición HTTP POST con el número del error correspondiente y un texto descriptivo separado por un espacio del código de error. En la siguiente tabla se listan los errores más comunes y sus códigos de error.

Código del error	Mensaje del error
900	Success
901	Insufficient memory
902	User cancelled
903	Loss of service
905	Attribute mismach
906	Invalid descriptor
951	Invalid DDVersion
952	Device aborted
953	Non-Acceptable content
954	Loader error

Tabla 0.1: Lista de los errores posibles en el proceso de descarga OMA

- Fase 1: Descarga del descriptor del contenido

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

El usuario después de navegar por las páginas que el servidor de presentación le provee escogerá un contenido a descargar. Cuando seleccione el enlace correspondiente se producirá una descarga HTTP, pero no del contenido, sino de su descriptor.

Como hemos visto el descriptor puede venir acompañado del propio contenido. En ese caso vendrá en una respuesta HTTP *multipart/related* en la que el descriptor vendrá primero y el contenido después. Las siguientes fases se ejecutarán igual con excepción de la número 5, descarga del contenido.

- Fase 2: Lanzamiento del agente de descargas y procesado del descriptor

En esta fase se lanza el programa en el cliente que gestiona las descargas y empieza su ejecución leyendo y procesando la información contenida en el descriptor. Cualquier error que se detecte en el descriptor parará la descarga y se notificará al servidor.

- Fase 3:Comprobación de capacidades del terminal

El terminal deberá comprobar si puede descargar e instalar el contenido. Chequeará si se dispone de suficiente memoria y si el formato es comprensible por el software instalado. En caso de que algo impida realizar la descarga con seguridad se notificará la cancelación de la misma indicando las causas.

En caso de problema, también es posible pasar a la siguiente fase para pedir al usuario confirmación indicando el problema encontrado.

- Fase 4: Confirmación del usuario

Una vez leído el descriptor del contenido y comprobado la utilidad del contenido, se debe preguntar al usuario si desea descargar el archivo indicándole por lo menos:

  - El nombre del contenido
  - El tamaño
  - La descripción
  - El proveedor
  - El tipo del contenido

- Fase 5: Descarga del contenido

Cuando el usuario de su conformidad, el terminal descargará el contenido mediante la dirección especificada en el descriptor del contenido. Cualquier error que se produzca durante dicha comunicación se deberá notificar.

- Fase 6: Instalación

Esta fase depende del tipo de contenido descargado. A grandes rasgos se puede decir que consiste en la preparación del contenido para su ejecución o visualización. Antes de mostrarlo o ejecutarlo, el terminal deberá enviar la notificación de éxito si procede.

- Fase 7: Notificación de la descarga

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

Mediante el parámetro de notificación del descriptor el terminal puede saber si es necesaria y en caso de serlo la dirección a la que hay que mandar el código con el estado y la descripción del mismo.

- Fase 8: Confirmación de la descarga y siguiente paso

Finalmente el terminal mostrará al usuario el resultado de la descarga. Además podrá permitir al usuario escoger una opción de continuar, que enlazará con la dirección especificada en el correspondiente parámetro del descriptor.

3.3.3 Descriptor

El descriptor es un archivo de texto en el que se incluye toda la información relativa al contenido y así como los datos necesarios para su descarga. Este punto es la principal diferencia entre la descarga de MIDLets y la descarga de contenidos tal y como define OMA.

En la descarga de MIDLets el descriptor es un archivo de texto en el que se listan las propiedades de forma sencilla, sin usar XML. Se denomina JAD (*Java Application Descriptor*).

Figura 36
Figura 37 MIDlet-Name: Show Properties MIDlet
Figura 38 MIDlet-Version: 1.0.1
Figura 39 MIDlet-Vendor: Core J2ME
Figura 40 MIDlet-Jar-URL: ShowProperties.jar
Figura 41 MIDlet-Jar-Size: 1190
Figura 42 MIDlet-1: ShowProps, , ShowProperties
Figura 43 MIDlet-Description: A simple property list example
Figura 44 JadFile-Version: 1.5
Figura 45 MIDlet-Data-Size: 500

En el caso del descriptor de *OMA Download* el archivo se basa en XML con las propiedades necesarias para la descarga. En este caso se le suele denominar DD (*Download Descriptor*) y se puede ver un ejemplo en el siguiente código:

Figura 46 <media xmlns="http://www.openmobilealliance.org/xmlns/dd">
Figura 47 <type>image/gif</type>
Figura 48 <objectURI>http:/download.example.com/image.gif</objectURI>
Figura 49 <size>100</size>
Figura 50 <installNotifyURI>http:/download.example.com/image.gif?id=image</installNotifyURI>
Figura 51 </media>

3.4 DRM

A partir de la experiencia de Internet y los PCs, en donde el pirateo de contenidos es la normal general, los operadores, proveedores y fabricantes de terminales han visto la necesidad de proveer mecanismos para la protección de los contenidos descargados.

Esta protección pasa por añadir en la descarga de los contenidos información en la que se especifiquen los permisos que ha comprado el usuario para el uso de ese contenido. Estos permisos indican si se

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

puede reproducir, guardar, copiar, las veces que se puede hacer o incluso el tiempo que se puede hacer. Evidentemente en el terminal, el sistema operativo tiene que proteger estos archivos a bajo nivel según esta información.

Debemos recordar en este momento que la descarga de contenidos se puede realizar bien mediante un mensaje MMS bien mediante algún tipo de conexión a Internet (WAP por ejemplo) utilizando el estándar *OMA Download*. Además, también soporta la transmisión de derechos para realizar *streamings* de contenidos multimedia.

3.4.1 Arquitectura

Como hemos visto la especificación DRM gira en torno al envío de un fichero que especifique los derechos que tiene el usuario para utilizar el contenido descargado. A este respecto hay tres posibles casos que corresponden con las tres arquitecturas definidas para DRM.

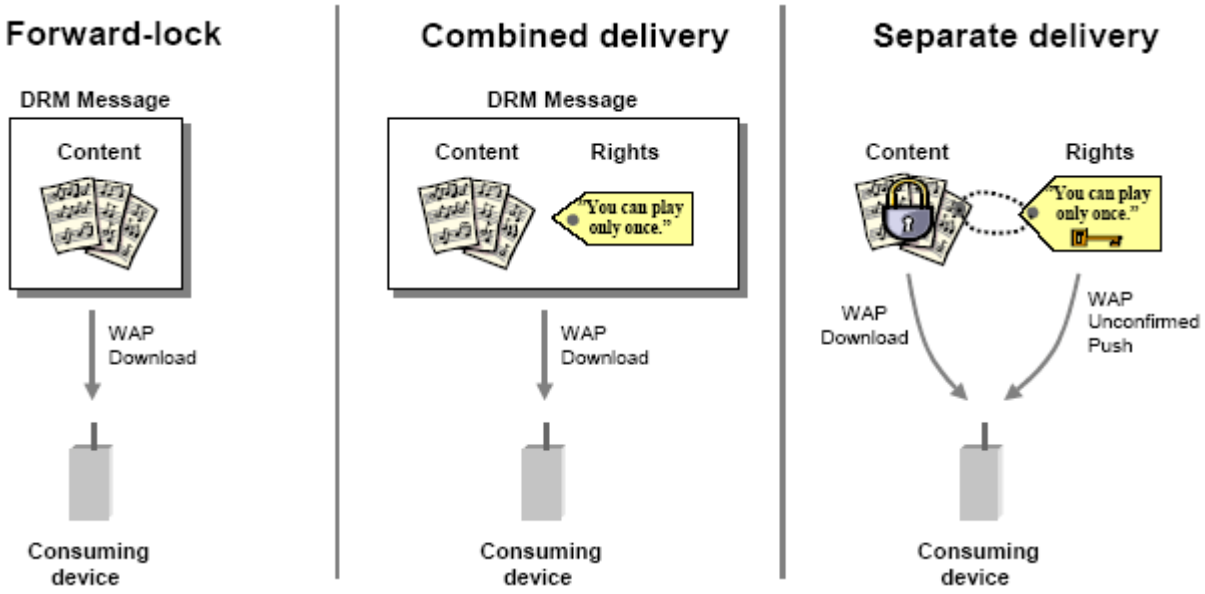

Figura 0.43: Diferentes escenarios de uso de la tecnología DRM Fuente: OMA

Las tres posibles arquitecturas se distinguen entre sí en la forma de mandar la información de los derechos. En la segunda se manda junto con el contenido, en la tercera se manda por separado y en la primera no envían derechos sino que se utilizarán unos por defecto.

- *Forward-lock*

  En este método el contenido se encapsula en un mensaje DRM y se envía al terminal sin incluir ningún derecho explícitamente. Como el nombre indica, el terminal podrá ejecutar, presentar, guardar el contenido al usuario pero no dejará reenviarlo a otro usuario ni sacarlo del terminal.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

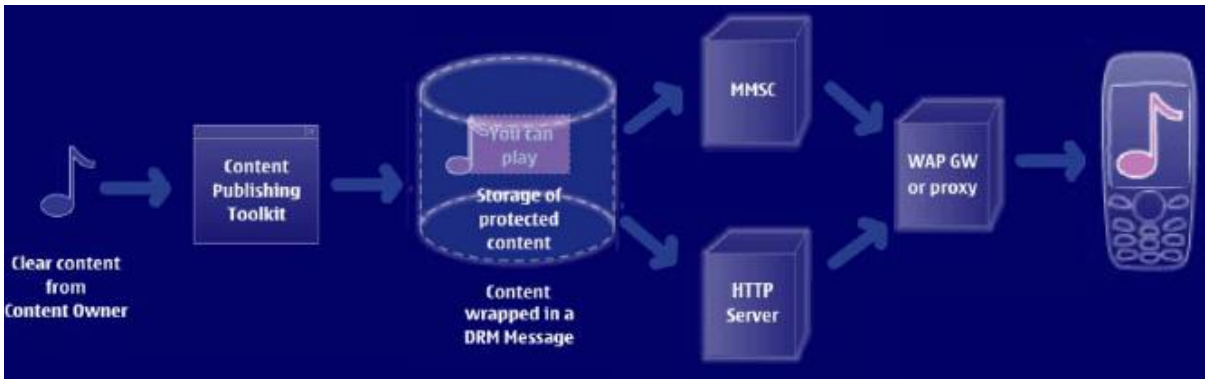

Figura 0.44: Caso de uso *Forward-Lock* Fuente: Nokia

- *Combined delivery*

En este método se especifican explícitamente los derechos de los que dispone el usuario para el uso del contenido. Estos derechos se expresan mediante un lenguaje específico y se envían en un mensaje DRM junto con el contenido.

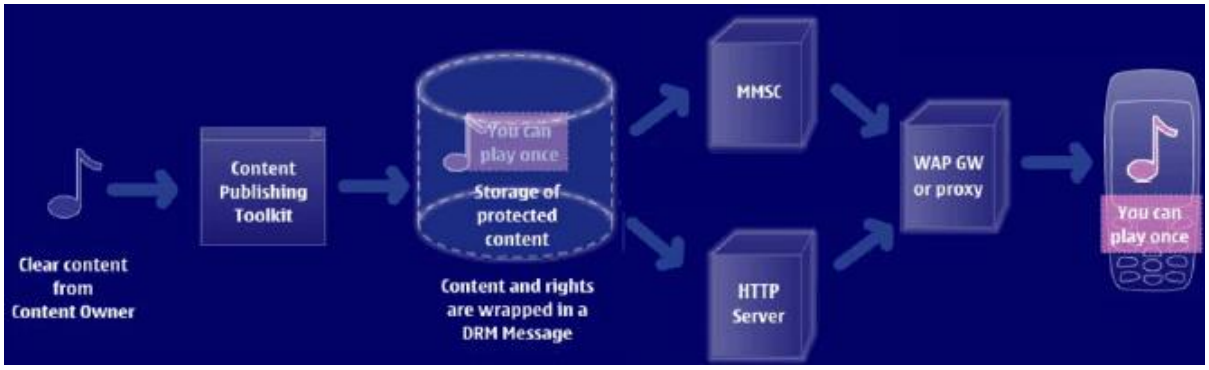

Figura 0.45: Caso de uso *Combined Delivery* Fuente: Nokia

- *Separate delivery*

Finalmente el tercer método consiste en enviar por separado los contenidos de los derechos. El contenido irá encapsulado en un mensaje DRM así como encriptados simétricamente. Posteriormente, se enviarán de algún modo (típicamente WAPPush) los derechos así como la clave necesaria para obtener los contenidos.

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

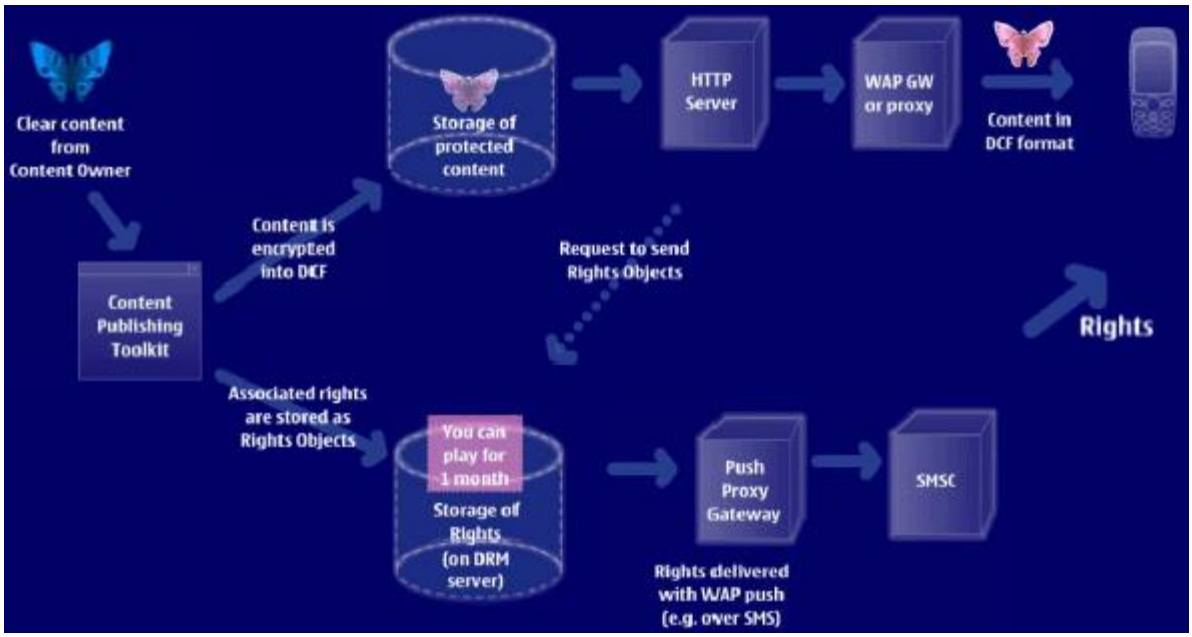

Figura 0.46: Caso de uso *Separate Delivery* Fuente: Nokia

3.4.2 Lenguaje de declaración de derechos

El lenguaje sobre el que se declaran los derechos se basa, como no podía ser de otra forma, en XML. Especifica lo que el terminal debe dejar hacer al usuario con el contenido descargado. Incluirá, por ejemplo, cuantas veces puede el usuario imprimir una imagen o durante cuanto tiempo puede el usuario escuchar una canción.

Figura 52 <o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX" xmlns:o-dd="http://odrl.net/1.1/ODRL-DD">
Figura 53 <o-ex:context>
Figura 54 <o-dd:version>1.0</o-dd:version>
Figura 55 </o-ex:context>
Figura 56 <o-ex:agreement>
Figura 57 <o-ex:asset>
Figura 58 <o-ex:context>
Figura 59 <o-dd:uid>cid:4567829547@foo.com</o-dd:uid>
Figura 60 </o-ex:context>
Figura 61 </o-ex:asset>
Figura 62 <o-ex:permission>
Figura 63 <o-dd:play/>
Figura 64 </o-ex:permission>
Figura 65 </o-ex:agreement>
Figura 66 </o-ex:rights>

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

Además en el caso de estar en la arquitectura en la que se mandan los derechos por separado, hay que mandar la clave con la que se podrán desencriptar los contenidos. Esta clave viene en el elemento del XML denominado *KeyValue*.

Figura 67 <o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX" xmlns:o-dd="http://odrl.net/1.1/ODRL-DD">
Figura 68 <o-ex:context>
Figura 69 <o-dd:version>1.0</o-dd:version>
Figura 70 </o-ex:context>
Figura 71 <o-ex:agreement>
Figura 72 <o-ex:asset>
Figura 73 <o-ex:context>
Figura 74 <o-dd:uid>cid:4567829547@foo.com</o-dd:uid>
Figura 75 </o-ex:context>
Figura 76 <ds:KeyInfo>
Figura 77 <ds:KeyValue>vUEwR8LzEJoeiC+dgT1mgg==</ds:KeyValue>
Figura 78 </ds:KeyInfo>
Figura 79 </o-ex:asset>
Figura 80 <o-ex:permission>
Figura 81 <o-dd:play/>
Figura 82 </o-ex:permission>
Figura 83 </o-ex:agreement>
Figura 84 </o-ex:rights>

3.4.3 Formato de archivo para los contenidos

El formato de los archivos que se envían al terminal se basa siempre en envíos *multipart* de HTTP. En el caso del método *forward-lock* y en el de *combined delivery* el envío se realiza sin encriptar y por lo tanto no hace falta mayor explicación, lo único a tener en cuenta es que en el caso de enviar derechos estos deben ir antes que el contenido en la estructura del mensaje.

En el caso del método *separated delivery* el contenido se encapsula en un formato en el que se incluyen varias cabeceras y posteriormente los datos encriptados.

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

Field name	Type	Purpose
Version	Uint8	Version number
ContentTypeLen	Uint8	Length of the ContentType field
ContentURILen	Uint8	Length of the ContentURI field
ContentType	ContentTypeLen octets	The MIME media type of the plaintext data.
ContentURI	ContentURILen octets	The unique identifier of this content object.
HeadersLen	Uintvar	Length of the Headers field
DataLen	Uintvar	Length of the Data field
Headers	HeadersLen octets	Headers define additional meta data about this content object.
Data	DataLen octets	The encrypted data

4. TRANSMISIÓN DE CONTENIDOS MULTIMEDIA

Una de las grandes aplicaciones de la transmisión de datos en el mundo móvil hace referencia a los contenidos multimedia. La importancia de estos contenidos no hace sino aumentar con el tiempo según van aumentando las capacidades de la red. De hecho se vuelven servicios necesarios para poder utilizar todo el ancho de banda de las redes 3G. Por este tipo de contenidos son de máxima actualidad y la estrella de las nuevas redes móviles celulares.

A la hora de hablar de transmisión de contenidos multimedia, para ser estrictos realmente podemos de tres tipos de transmisiones:

- Transmisiones con envíos de mensajes multimedia con los contenidos.

  El nuevo estándar MMS permite mandar todo tipo de contenidos (incluso video) mediante un mensaje multimedia. Evidentemente es un servicio diferido y, puesto que se envía todo el contenido y luego se visualiza, el tamaño de los contenidos multimedia no puede ser muy grande. Además, los mensajes son gestionados y almacenados para su envío (arquitectura *Store&Forward*) en el centro de mensajes (MMSC) por lo que también limitará la capacidad de éste el tamaño máximo del mensaje. Normalmente, como estos centros tienen que gestionar cantidades ingentes de mensajes no suelen permitir un tamaño muy alto (200 Kb) para no saturarse.

- Transmisiones mediante descargas

  En el anterior apartado hemos estado hablando de las distintas formas de descarga de contenidos y cómo funcionan. En resumen, los usuarios mediante la navegación por distintas páginas residentes en servidores seleccionarán un contenido a descargar y lo recibirán mediante una transacción estándar. Al igual que en la mensajería, la visualización del contenido se realiza después de descargarlo entero por lo que tampoco el contenido puede ser muy grande. Aún así, no tiene la limitación de estar en medio un centro de mensajes por lo que aún siendo pequeño el tamaño máximo razonable que se puede descargar siempre podría ser mayor que el de un MMS.

- Transmisiones mediante *streaming*

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

A grandes rasgos, los servicios de transmisión de contenidos multimedia mediante *streaming* consisten en el envío de los contenidos multimedia como un flujo continuo de datos y la reproducción de los mismos en el terminal a la vez que se reciben y sin tener que almacenarlos en el mismo. De esta forma el tamaño de los contenidos transmitidos pueden ser realmente más grandes puesto que no hace falta recibirlos enteros para empezar la reproducción y los datos del contenido multimedia no tienen porque ser almacenados en el terminal. Este tipo de trasmisiones está claramente orientado a la emisión de contenidos de audio y video.

Las diferencias entre este sistema y las descargas son claras. Por poner un ejemplo, en la televisión los datos recibidos y reproducidos simultáneamente se pueden considerar *streaming*, mientras que la analogía con las descargas podrían ser el visionado de una película de vídeo que ya está almacenada antes de reproducirla.

Pero no todo son ventajas, puesto que se reproduce a la vez que se van recibiendo los datos necesarios para seguir con la reproducción, es necesario que el caudal que se recibe se mantenga constante o no sufra caídas bruscas, por lo que se puede decir que los requisitos son de tiempo real, si no llegan los siguientes datos la reproducción se parará con la consiguiente falta de calidad y mala experiencia para el cliente

.

A lo largo de este apartado vamos a ver los principales puntos a estudiar en la transmisión de contenidos multimedia, como los formatos aceptados por el grupo 3GPP para las redes móviles o los protocolos necesarios para realizar la transmisión de los contenidos. Finalmente, analizaremos la adaptación de contenidos, cuestión importante en un entorno en la que la variedad en las capacidades de los terminales es tal que se han definido estándares para establecer las características de los dispositivos.

Puesto que ya hemos visto anteriormente tanto la mensajería multimedia como las descargas, vamos a centrarnos en el análisis del *streaming* de contenidos multimedia en los posteriores apartados. Aún así, muchas cuestiones como la adaptación de contenidos o los formatos de los mismos son cuestiones generales que afectan tanto a los MMS, como las descargas, así como incluso a la creación de páginas WAP o WEB para terminales móviles.

Muchos de los puntos que vamos a ver están todavía en proceso de estandarización y evolución. Por ejemplo, ya se está pensando en proporcionar *streaming* con los mensajes multimedia y también se empieza a estandarizar la gestión de derechos de los contenidos transmitidos mediante *streaming* (DRM).

4.1. Formatos

Los formatos disponibles actualmente para los contenidos en Internet son realmente muy numerosos. Sólo hace falta una breve navegación por Internet para darnos cuenta la cantidad de formatos de sonido, imágenes y vídeos que existen. Debido a esta gran variedad y a las restricciones debidas a las capacidades limitadas de los terminales, el grupo 3GPP ha definido los formatos que se recomienda implementar en los terminales. De esa forma los proveedores de contenidos y aplicaciones pueden

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

saber qué formatos deben utilizar para codificar sus contenidos y que puedan ser utilizados por la mayoría de terminales.

En la siguiente tabla se ilustran los formatos y *codecs* recomendados por el 3GPP.

Tipo de contenido	**Formato o Codecs**
Voz	AMR
Audio	MPEG-4 AAC Low Complexity MPEG-4 AAC Long Term Prediction (opcional)
Audio sintetizado	Midi
Video	H.263 MPEG4 (opcional)
Imágenes	JPEG
Gráficos	PNG, GIF
Gráficos vectoriales	SVG
Texto	XHTML MP SMIL

4.2. Protocolos

Respecto a los protocolos para transmitir contenidos multimedia, primero debemos recordar que ya hemos estudiado la mayoría cuando analizamos el funcionamiento de MMS, WAP o las descargas. Aún así, nos queda por ver un importante conjunto que hace referencia al *streaming* de contenidos.

El servicio de *streaming* de contenidos multimedia ha sido especificado por el grupo 3GPP basándose en estándares de Internet como los protocolos RTP (*Real Time Transport Protocol*), RTCP (*Real Time Control Protocol*), RTSP (*Real Time Streaming Protocol*) y SDP (*Session Description Protocol*). En las siguientes páginas, vamos a analizar las principales funciones y características de cada uno de ellos.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

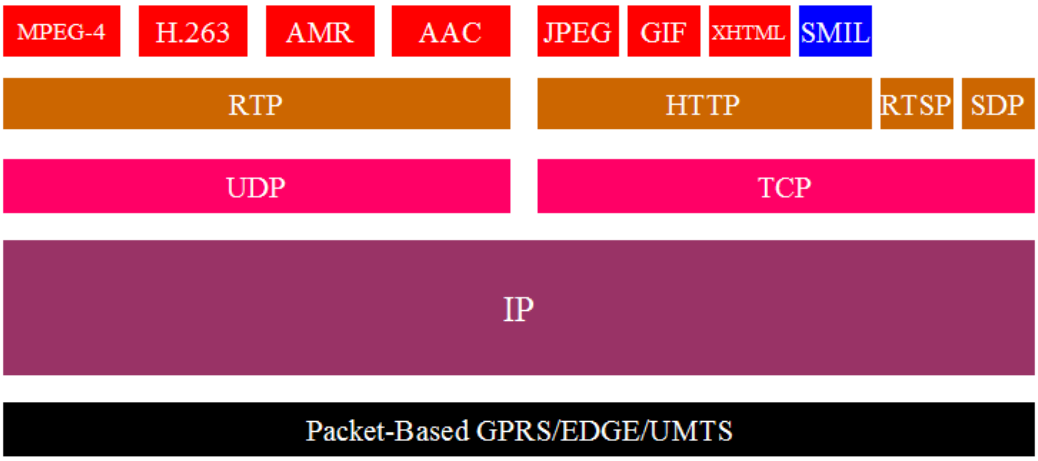

Figura 0.47: Pila de protocolos en la transmisión de contenidos multimedia Fuente Emblaze Research

4.2.1. RTP

El protocolo de transporte en tiempo real RTP es un protocolo IP que proporciona soporte para la transmisión de datos en tiempo real, como por ejemplo *streams* de contenidos multimedia (video y audio). Las funciones que RTP proporciona incluyen la reconstrucciones de temporizaciones, la detección de pérdida de paquetes y la seguridad e identificación de los contenidos transmitidos.

Aunque RTP ha sido diseñado para funcionar en modo *multicast* también puede hacerlo en modo *unicast*. Se puede usar igualmente para transporte unidireccional, como video bajo demanda, o para servicios interactivos como la telefonía por Internet.

RTP trabaja conjuntamente con el protocolo auxiliar de control que vamos a ver a continuación, y que sirve para obtener realimentación de la calidad de la transmisión así como para obtener información sobre los participantes de la sesión.

Puesto que se basa en IP, funciona sobre una red que transmite datagramas por lo que los paquetes enviados se reciben con un retardo no predecible y en algunos casos desordenados (si trabajamos encima de UDP), sin embargo las aplicaciones multimedia requieren temporizaciones adecuadas en la transmisión de datos y su reproducción. RTP proporciona marcas temporales, numeración de secuencias y otros mecanismos para proporcionar un soporte robusto a la temporización. Con estos mecanismos, RTP provee un transporte punto a punto en tiempo real sobre la red basada en datagramas o paquetes.

La temporización es la información más importante en las aplicaciones de tiempo real, no sólo porque haya que saber qué datos son los que hay que utilizar en cada momento, sino también para sincronizar los diferentes contenidos que pueden estar llegando simultáneamente (por ejemplo, audio y video).

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

Quien envía asigna una marca temporal (*timestamp*), que corresponde al momento en el que el primer *byte* del paquete fue muestreado. El receptor utilizará esta marca temporal para reconstruir la temporización y reproducir los datos en el momento y a la velocidad correctos. Como hemos dicho, antes la temporización también es importante para sincronizar distintas secuencias de datos recibidas simultáneamente, aunque RTP no es responsable de esta sincronización, sino que es realiza por los protocolos superiores de aplicación.

Los dos protocolos de transporte más usados sobre IP son TCP y UDP. TCP proporciona un flujo fiable y orientado a conexión entre dos terminales, mientras que UDP provee un servicio de datagramas no fiable (pueden llegar datos repetidos, desordenados o incluso no llegar) y no orientado a conexión. Aunque RTP se puede montar encima de TCP, está pensado para funcionar sobre todo encima de UDP. Esta elección se debe principalmente a que RTP ha sido diseñado para datos en tiempo real y por lo tanto la fiabilidad del transporte no es tan importante como la recepción en el tiempo adecuado. Es más, TCP implementa la fiabilidad de la comunicación mediante retransmisiones de paquetes IP, lo cual en el caso de tráfico de datos en tiempo real es un inconveniente. En el caso de congestión de la red, los paquetes perdidos, simplemente, redundarán en una calidad inferior, pero aceptable, en cambio, si el protocolo insiste en la transmisión fiable, retransmitiendo paquetes perdidos, el retardo posiblemente aumentará, degradando más la calidad de la recepción.

Puesto que UDP no despacha los paquetes ordenados en el tiempo, se usa la numeración de secuencias para, en recepción, ordenar los paquetes recibidos. La numeración de secuencias se usa también para la detección de pérdidas de paquetes. En la práctica, RTP se implementa generalmente dentro de la aplicación, pues temas como la recuperación de paquetes perdidos o de control de las congestiones de red, tienen que ser implementados al nivel de aplicación.

Otra función de RTP es identificar el formato de los datos así como el codec con el que se han comprimido los mismos. Para realizar esta labor utiliza un identificador de carga *payload*, con el que la aplicación receptora sabe como interpretar y reproducir los datos. Además, el protocolo RTP permite la identificación de las fuentes de los datos. Permite a la aplicación receptora conocer de dónde vienen los datos, por ejemplo, en una audio conferencia, a partir del identificador de la fuente se puede saber quién está hablando.

Para establecer una sesión RTP, la aplicación define una pareja de direcciones destino de transporte (una dirección de red con un par de puertos para RTP y RTCP). En una sesión de transmisión de datos multimedia, cada tipo de datos es transportado por una sesión separada de RTP, con sus propios paquetes de RTCP, que informan de la calidad de recepción para esa sesión. A continuación vamos a ver más en profundidad el protocolo auxiliar de control para RTP, el RTCP.

4.2.2. RTCP

RTCP (*Real Time Control Protocol*) es un protocolo diseñado para trabajar conjuntamente con RTP. En una sesión RTP, los participantes se envían cada cierto tiempo paquetes RTCP para tener información sobre la calidad de la recepción. Se definen cinco tipos de paquetes RTCP para transportar la información de control, a saber:

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

- *RR (*informe de receptor). Son los informes enviados por los participantes que actúan como receptores. Estos informes contienen datos acerca de la calidad de la recepción, incluyendo el número más alto de secuencia recibido, el número de paquetes perdidos, la información sobre paquetes desordenados y las marcas temporales para calcular el retardo de ida y vuelta entre receptor y el transmisor.
- *SR* (informe de emisor). Los informes de emisor son generados por los emisores activos. Además de datos sobre la calidad de la recepción, como en los RR, contienen datos de sincronización, de contadores acumulativos de paquetes y del número de *bytes* enviados.
- *SDES* (datos de descripción de fuente*).* Contienen información descriptiva de la fuente de datos.
- *BYE*. Indica el fin de la participación.
- *APP* (datos específicos de la aplicación). Se han reservado para usos experimentales, mientras se desarrollan nuevas funciones y tipos de aplicaciones.

A través de estos paquetes, RTCP puede proporcionar las funciones para las que ha sido diseñado y que a continuación se detallan:

- Monitorización de la calidad de servicio (QoS) y congestión de red

La función primaria de RTCP es proporcionar realimentación a una aplicación sobre la calidad de la distribución de los contenidos. Esta información es útil a los emisores, a los receptores y a otras terceras partes interesadas (monitores de aplicación, por ejemplo). El emisor puede ajustar su transmisión basándose en estos informes. El receptor puede determinar si la congestión de red es local, regional o global.

- Identificación de las fuentes

Las fuentes de datos se identifican en los paquetes RTP con identificadores de 32 bits generados aleatoriamente. Estos identificadores no son apropiados para usuarios humanos. Los paquetes SDES contienen identificadores únicos globales e información textual, como el nombre de los participantes, el número de teléfono, la dirección de e-mail, etc.

- Sincronización intermedia

RTCP envía informes con información de tiempo real que corresponde con una determinada marca temporal RTP. Esa información puede ser utilizada para sincronizar fuentes de datos que procedan de distintas sesiones RTP.

- Escalado de la información de control

Los paquetes RTCP se envían periódicamente entre los participantes. Cuando el número de participantes aumenta, se hace necesario establecer un compromiso entre la obtención de información actualizada y la sobrecarga por tráfico de red. RTP limita el tráfico de control al 5 por ciento del tráfico total de la sesión, esto se consigue ajustando el tráfico RTCP a un régimen acorde al número de participantes.

4.2.3. RTSP

El concepto de *streaming* se basa en que en vez de almacenar grandes ficheros multimedia y reproducirlos, los datos multimedia se envían a través de la red secuencialmente en *streams*. El proceso de

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

streaming rompe los datos en paquetes del tamaño apropiado para su transmisión entre servidor y cliente. Un cliente puede reproducir el primer paquete, mientras decodifica el segundo y recibe el tercero. Así, el usuario puede disfrutar de los contenidos sin esperar a que finalice la transmisión.

RTSP (*Real Time Streaming Protocol*) es el protocolo de *streaming* en tiempo real, un protocolo multimedia cliente-servidor de presentación para permitir el despacho controlado de los datos multimedia a través de la red IP. Proporciona una interfaz, al estilo de un reproductor de vídeo, con funciones de control remoto como "pausa", "avance rápido", "atrás", e "ir a posición".

RTSP es un protocolo a nivel de aplicación, diseñado para trabajar conjuntamente con protocolos de bajo nivel como RTP. Proporciona mecanismos para seleccionar canales de envío (como UDP, UDP *multicast*, y TCP). Se puede ver RTSP como un control remoto en red entre el servidor y el cliente.

RTSP pretende proporcionar, para presentaciones multimedia, los mismos servicios que HTTP proporciona para textos y gráficos, de hecho se ha diseñado intencionadamente con una sintaxis similar, de tal modo que la mayor parte de los mecanismos de extensión de HTTP se pueden añadir a RTSP.

En RTSP cada presentación y cada *stream* multimedia es identificado por una URL RTSP. La presentación completa y las propiedades de los medios se definen en un fichero de descripción de presentación, entre la información se incluye el tipo de codificación, el idioma, las URLs RTSP, las direcciones de destino, los puertos y otros parámetros. El cliente puede obtener este fichero mediante HTTP, *e-mail* u otros métodos.

Aún así, RTSP difiere de HTTP en muchos aspectos. En primer lugar, HTTP es un protocolo sin estados y RTSP ha de mantener los estados de la sesión para enlazar las peticiones con los *streams* relacionados. En segundo lugar, HTTP es asimétrico, cuando el cliente realiza una petición el servidor responde, mientras que en RTSP ambos, cliente y servidor, pueden realizar peticiones.

Los servicios y operaciones soportados son:

- *Options*. El cliente o el servidor comunican a la otra parte las opciones que pueden aceptar.
- *Describe*. El cliente recupera la descripción de una presentación o medio identificado por una URL.
- *Announce*. Cuando se envía desde el cliente al servidor, *Announce* envía la descripción de una presentación identificada por su URL. Cuando se envía desde el servidor al cliente, *Announce* actualiza la descripción de sesión en tiempo real.
- *Setup*. El cliente le pide al servidor que reserve recursos para un *stream* y para iniciar una sesión RTP.
- *Play*. El cliente le pide al servidor que comience a enviar datos para el *stream* reservado, vía *Setup*.
- *Pause*. El cliente, temporalmente, para el flujo del *stream* sin liberar los recursos asociados.
- *Teardown*. El cliente le pide al servidor que detenga el envío de un determinado *stream* y libere los recursos asociados.
- *Get_Parameter*. Recupera el valor de un parámetro de una presentación o un *stream* identificado por su URI.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

- *Set_Parameter*. Asigna el valor de un parámetro de una presentación o *stream* identificado por su URL.
- *Redirect*. El servidor informa al cliente de que se debe conectar a otra dirección de servidor. La cabecera obligatoria de localización indica el URL que el cliente debe contactar.
- *Record*. El cliente inicia la grabación de unos datos multimedia de acuerdo a la descripción de la presentación.

Algunos de estos métodos pueden ser enviados tanto por el cliente como por el servidor, pero otros sólo pueden ser emitidos en una dirección. No todos los métodos son necesarios para tener un servidor plenamente funcional. Por ejemplo, un servidor de medios, alimentado con datos en vivo, puede no soportar el método *Pause*.

Las solicitudes RTSP se envían, como normal general, por un canal independiente del canal de datos. Pueden ser transportadas por conexiones persistentes o creando una conexión por petición-respuesta.

4.2.4. SDP

El protocolo SDP (*Session Description Protocol*) es una sintaxis de descripción de la sesión basada en texto, que informa de la dirección del servidor, de los contenidos de los *streams*, y *codecs* necesarios y de los contenidos solicitados durante la sesión por el cliente.

Seguramente, la mejor comparación que podamos hacer sobre los archivos SDP sean los descriptores de descargas que veíamos en el capítulo anterior. Al fin y al cabo el objetivo es el mismo, mandar información al cliente receptor sobre el contenido multimedia que se va a transmitir y cómo se va a hacer tal y como podemos ver en el siguiente ejemplo:

Figura 85 v=0
o=ghost 2890844526 2890842807 IN IP4 192.168.10.10
s=3GPP Unicast SDP Example
i=Example of Unicast SDP file
u=http://www.infoserver.com/ae600
e=ghost@mailserver.com
c=IN IP4 0.0.0.0
t=0 0
Figura 86 a=range:npt=0-45.678
m=video 1024 RTP/AVP 96
Figura 87 b=AS:128
a=rtpmap:96 H263-2000/90000
a=fmtp:96 profile=3;level=10
a=control:rtsp://mediaserver.com/movie.3gp/trackID=1
a=framesize:96 176-144
a=recvonly

4.3. Adaptación de contenidos

En la transmisión de contenidos multimedia en el entorno móvil hay un apartado que nunca debemos olvidar. Los terminales móviles tienen grandes diferencias entre sí y además existen gran cantidad de modelos diferentes. Esto provoca que los contenidos deban ser adaptados en la medida de lo posible

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

al terminal receptor. Unas veces habrá que recodificar los contenidos cambiándoles el formato a uno que el dispositivo entienda, otras veces habrá que cambiar propiedades del contenido como altura y anchura, e incluso otras no habrá que listar el contenido al usuario porque no lo va a soportar su dispositivo.

Una de las decisiones más importantes cuando hablamos de adaptar contenidos es el coste computacional de realizar la codificación de contenidos multimedia en tiempo real. Si creemos que puede ser demasiado alto para nuestras expectativas de tráfico y nuestras capacidades hardware deberemos plantearnos soluciones basadas en agrupar los terminales en familias y realizar la adaptación en tiempo de *deployment* una sola vez.

El mayor problema que tienen las aplicaciones multimedia que no incluyen la adaptación de contenidos al terminal es que tienden a codificar los contenidos para el caso peor, es decir para que funcione por lo menos en el terminal con peores características. Esto lleva a que en terminales avanzados se vean los contenidos de forma muy pobre. Un ejemplo característico de este problema son las imágenes WAP pequeñas y en blanco y negro en terminales con pantallas grandes y en color. La sensación que dan estas páginas es muy perjudicial y es una de las causas del fracaso de WAP.

Además, otro motivo para adaptar los contenidos al terminal es optimizar el ancho de banda usado. Por ejemplo, no tiene sentido mandar una imagen a un terminal mayor que su pantalla puesto que el terminal o bien la descartará o bien la reducirá. El primer caso es nefasto, pero el segundo tampoco es óptimo porque hemos gastado dinero y ancho de banda innecesarios.

La industria enseguida ha sido consciente de esta problemática y ha optado por darle soluciones por dos vías. Por un lado ha optado por estandarizar los formatos de los contenidos multimedia de tal forma que casi todos los terminales nuevos soportan los mismos formatos. Además se han diseñado mecanismos para que el terminal pueda comprobar si soporta el contenido multimedia antes de descargarlo (aunque lo óptimo sería no mostrar los contenidos que el usuario no pudiese utilizar).

La segunda solución pensada es la comunicación de todas las características relevantes del terminal a la aplicación para que pueda particularizar y adaptar los contenidos a éstas. Para conseguir este objetivo se ha diseñado la tecnología UAProf (*User Agent Profile*) que permite obtener un fichero XML con toda la información necesaria sobre el terminal.

En los posteriores apartados vamos a analizar primero el estándar UAProf, para luego entrar en las distintas posibilidades de adaptación que se pueden realizar para los tipos más importantes de contenidos multimedia, las imágenes, los sonidos y los vídeos.

4.3.1. UAProf

El estándar UAProf (*User Agent Profile*) define las transacciones que se deben realizar entre el terminal, los proxies intermedios y el servidor final para poder comunicar con éxito las capacidades del terminal. Esta tecnología, también conocida como CPI (*Capability and Preference Information*) usa el modelo emergente en Internet llamado CC/PP (*Composite Capability/Preference Profile*).

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

La definición de la información enviada referente a las características del terminal engloba las siguientes áreas entre otras menos importantes:

- Características del hardware del terminal (pantalla, teclas, fabricante)
- Características del software del terminal (sistema operativo, codecs de audio y video, soporte para aplicaciones nativas)
- Preferencias del usuario y de las aplicaciones (versión y modelo del navegador, lenguajes de marcado soportados, etc.)
- Características WAP (versión de WAP, librerías WMLScript, tamaño máximo de WML, etc.)

Todas estas características son incluidas en un archivo XML-RDF que puede ser analizado para personalizar las aplicaciones móviles. En la siguiente figura se puede ver un ejemplo de un fichero de un terminal real.

```
<?xml version=”1.0”?>
<!DOCTYPE rdf:RDF [
<!ENTITY ns-rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
<!ENTITY ns-prf 'http://www.openmobilealliance.org/tech/profiles/UAPROF/ccppschema-
YYYYMMDD#'>
<!ENTITY prf-dt 'http://www.openmobilealliance.org/tech/profiles/UAPROF/xmlschema-
YYYYMMDD#'>
]>
<rdf:RDF xmlns:rdf=”&ns-rdf;”
xmlns:prf=”&ns-prf;”>
<rdf:Description rdf:ID=”MyDeviceProfile”>
  <prf:component>
    <rdf:Description rdf:ID=”HardwarePlatform”>
      <rdf:type rdf:resource=”&ns-prf;HardwarePlatform”/>
      <prf:ScreenSizeChar rdf:datatype=”&prf-dt;Dimension”>15x6</prf:ScreenSizeChar>
      <prf:BitsPerPixel rdf:datatype=”&prf-dt;Number”>2</prf:BitsPerPixel>
      <prf:ColorCapable rdf:datatype=”&prf-dt;Boolean”>No</prf:ColorCapable>
      <prf:TextInputCapable rdf:datatype=”&prf-dt;Boolean”>Yes</prf:TextInputCapable>
      <prf:ImageCapable rdf:datatype=”&prf-dt;Boolean”>Yes</prf:ImageCapable>
      <prf:Keyboard rdf:datatype=”&prf-dt;Literal”>PhoneKeypad</prf:Keyboard>
      <prf:NumberOfSoftKeys rdf:datatype=”&prf-dt;Number”>0</prf:NumberOfSoftKeys>
    </rdf:Description>
  </prf:component>
  <prf:component>
    <rdf:Description rdf:ID=”SoftwarePlatform”>
      <rdf:type rdf:resource=”&ns-prf;SoftwarePlatform”/>
      <prf:AcceptDownloadableSoftware rdf:datatype=”&prfdt;Boolean”>No</prf:Ac-
ceptDownloadableSoftware>
      <prf:CcppAccept-Charset>
        <rdf:Bag>
          <rdf:li rdf:datatype=”&prf-dt;Literal”>US-ASCII</rdf:li>
          <rdf:li rdf:datatype=”&prf-dt;Literal”>ISO-8859-1</rdf:li>
          <rdf:li rdf:datatype=”&prf-dt;Literal”>UTF-8</rdf:li>
```

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

Figura 120 <rdf:li rdf:datatype=”&prf-dt;Literal”>ISO-10646-UCS-2</rdf:li>
Figura 121 </rdf:Bag>
Figura 122 </prf:CcppAccept-Charset>
Figura 123 </rdf:Description>
Figura 124 </prf:component>
Figura 125 </rdf:Description>
Figura 126 </rdf:RDF>

De forma resumida, se puede decir que en las peticiones que realizan los clientes, se envía la cabecera *x-wap-profile* con la URL del fichero XML con las características. En realidad la situación es más compleja y a las peticiones que envían los terminales los elementos de red intermedios pueden añadir modificaciones.

4.3.2. Imágenes

Una imagen almacenada en un ordenador puede ser abstraída por una matriz bidimensional de puntos. Cada punto es denominado píxel y son muestras discretas de una imagen real en un espacio continuo. Estos píxeles tienen almacenado el color que se debe ver en este punto de la imagen.

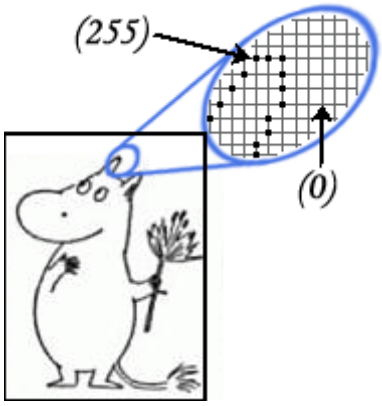

Figura 0.48: Imagen como matriz de píxel

Para almacenar la información de la imagen en tipos de datos conocidos por el ordenador, la imagen se estructura en bandas. Cada banda es una matriz bidimensional que almacena la información de color en una de las dimensiones de los espacios de colores. Por ejemplo, uno de los formatos de color más utilizados es el RGB (Red Green Blue), una imagen almacenada de esta forma poseerá tres bandas. Cada banda contendrá información del valor de cada píxeles en cada color. Si se desea tener una imagen en grises, solamente hará falta una banda para almacenar la información.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

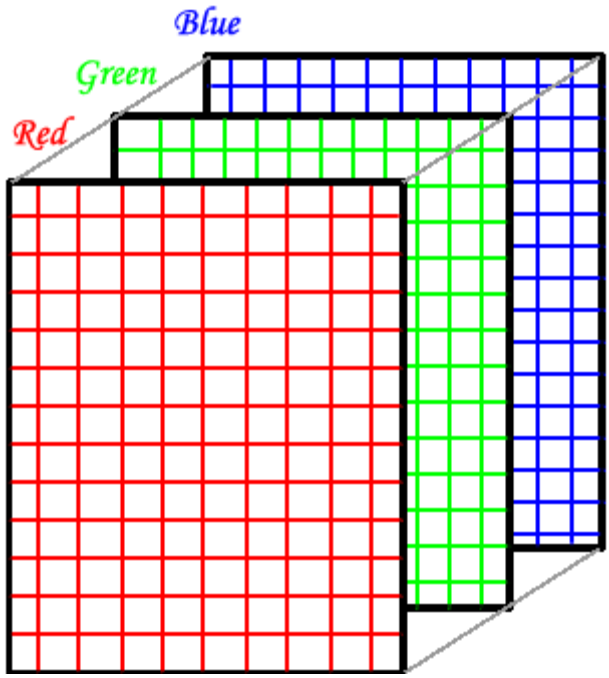

Figura 0.49: Imagen RGB

Para almacenar el valor de cada píxel en cada banda necesitaremos un tipo primitivo como entero, byte o entero largo. No son necesarios los decimales y según la precisión que queramos utilizaremos uno u otro tipo. Esta decisión tendrá como consecuencia una mayor gama de colores disponibles para presentar la imagen. Tal y como hemos descrito una imagen, podría ser almacenada como una matriz tridimensional de enteros o algún tipo de coma fija.

Una vez que hemos visto como se almacena una imagen podemos realizar unos sencillos cálculos que nos demostrarán el gran espacio que ocupan si no se comprimen de alguna forma. Si tenemos una imagen a color de 1000x1000 píxeles (tamaño normal) y se almacena con una profundidad de color de 24 bits (4 bandas de 1 byte), obtenemos:
1000x1000x4= 4 Mbytes.

Un tamaño que es claramente excesivo. Para guardar las imágenes en archivos almacenables en discos o para trasmitirlas por red de forma eficiente se utilizan formatos gráficos especiales. Estos formatos se aprovechan de características estadísticas o del ojo humano para reducir el tamaño que ocupa la imagen en el archivo. Entre estos archivos que comprimen la imagen podemos encontrar JPEG, GIF o PNG.

En las aplicaciones de Internet ya estamos acostumbrados a ver sin problemas todo tipo de imágenes. Pero en el mundo móvil las cosas no son tan fáciles. Como veremos hace falta adaptar la imagen para el terminal que la quiere visualizar o descargar. Las características que se pueden y deben adaptar de una imagen son las siguientes:

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

- El formato

No todos los terminales aceptan todos los formatos. Por esta razón hay que cambiar el formato de la imagen si el terminal no lo soporta. Los 4909 más normales que soportan los terminales son WBMP, JPEG, GIF y PNG.

- Las dimensiones

Muchos terminales no muestran las imágenes que reciben si éstas son más grandes que la pantalla. Otros reducen la imagen para adaptarla. Ambas situaciones no son deseables, la primera por que la imagen no se ve. La segunda porque supone un desperdicio de ancho de banda. Aún así, a veces el terminal permite realizar *scroll* vertical u horizontal, en estos casos es discutible la reducción de la dimensión correspondiente.

Otra razón para reducir las dimensiones de la imagen es que el tamaño máximo de fichero que soporta el terminal sea menor que el tamaño de la imagen. Muchos terminales tienen limitaciones respecto al número de bytes que se pueden descargar en un mismo fichero.

- Color/Grises

Evidentemente, si una imagen es color y el terminal tiene una pantalla en blanco y negro o con escala de grises habrá que convertir la imagen original.

- Profundidad de color

Finalmente también puede darse el caso de que la imagen posea más colores/grises de los que soporta el terminal. En ese caso hay que reducir los colores/grises. Para realizar esta labor la técnica más avanzada (sobre todo con fotografías) es el *Dithering*. Mediante la utilización de transformaciones basadas en matrices se puede reducir el número de colores con bastante calidad.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

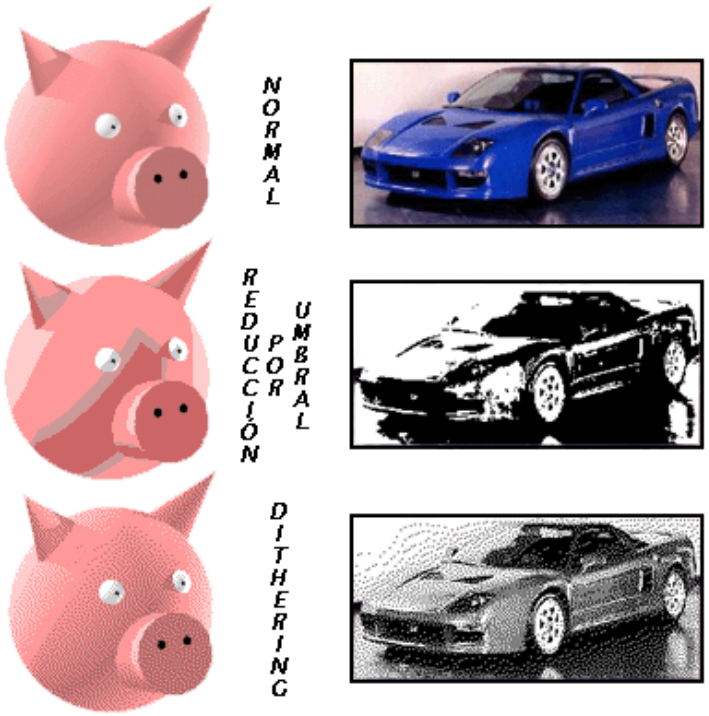

Figura 0.50: Ejemplo del uso de la técnica *Dithering*

En el caso de pasar imágenes a blanco y negro (pantallas muy comunes en los terminales antiguos), obtendremos mejores resultados si además combinamos el *Dithering* con algoritmos de ecualización del histograma y mejora del contraste.

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

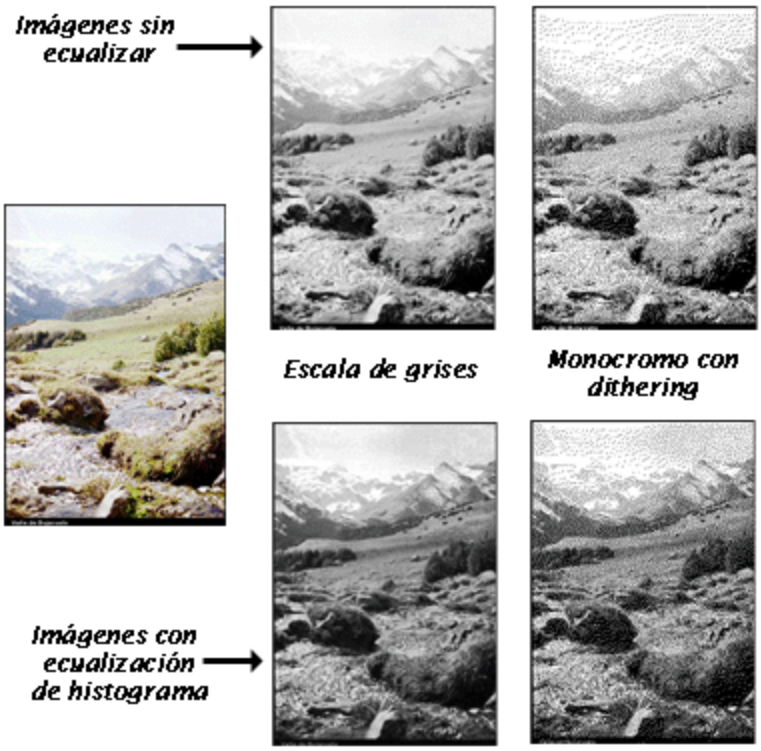

Figura 0.51: Diferentes algoritmos para obtener imágenes monocromo

Otro motivo para reducir la profundidad de color de una imagen es reducir el tamaño del fichero a transmitir con objeto de optimizarlo o reducirlo a menos del máximo del terminal.

4.3.3. Sonidos

Los sonidos son un contenido multimedia totalmente diferente a las imágenes. Para empezar hay dos grandes formas de representar sonidos en el mundo digital. La primera forma se suele denominar como sonido muestreado o "sampleado" (*sampled*). Las muestras (*samples*) son sucesivas fotos instantáneas de la señal. La onda sonora es convertida por los micrófonos en una señal eléctrica analógica, de la cual se extraen muestras a una determinada velocidad y se convierte a digital.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

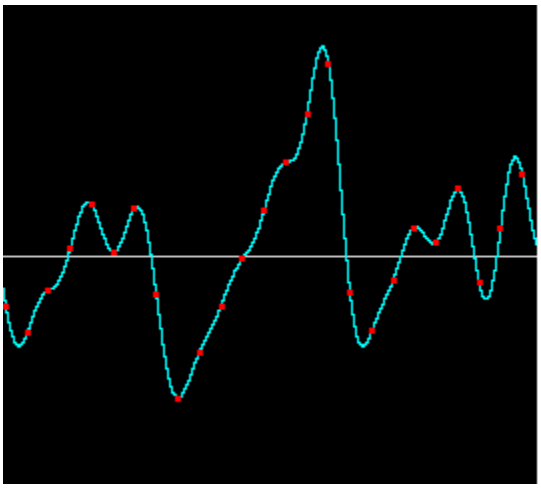

Figura 0.52: Sonido muestreado

Este gráfico representa la amplitud de onda analógica sonora en el eje vertical y el tiempo en el eje horizontal. La amplitud de la onda es medida periódicamente según cierta frecuencia, y convertida a digital mediante el proceso de cuantificación (se redondean los valores continuos a un cierto número de valores discretos). La precisión y calidad del sonido digital dependerán de la resolución temporal (o frecuencia de muestreo, velocidad con la que se obtienen las muestras) y de la resolución en amplitud (o cuantificación, el número de bits que sirven para digitalizar cada muestra. Como referencia, la música almacenada en los CDs de música se muestrea a 44.100 muestras por segundo y se cuantifica a 16 bits por muestra. Si echamos las cuentas veremos que si no comprimimos de alguna forma los sonidos, las muestras digitalizadas ocuparán un gran espacio en el disco o en el canal de transmisión.

La segunda forma de almacenar un sonido es guardar la forma en la que se debe reproducir. Básicamente se almacenan eventos que deben ser reproducidos por un sintetizador. Es como guardar las partituras de los instrumentos que van a tocar la melodía.

Por poner una analogía comparando ambas formas de almacenamiento y codificación, los sonidos muestreados podrían verse como una imagen escaneada, mientras que un sonido que se va a sintetizar podría verse como una imagen vectorial que ha de ser interpretada para poder visualizarla.

Los sonidos son más difíciles de adaptar que las imágenes. Al final lo único que es viable adaptar son los formatos de los sonidos. Si el terminal no soporta la codificación usada hay que cambiarla. Los formatos de sonidos más comunes en los terminales son AMR, WAV (sonido muestreado sin compresión), MIDI (sonido que se ha de sintetizar), MP3, AAC, i-Melody, etc.. Para añadir dificultad, no siempre son posibles todas las transformaciones. Por ejemplo de WAV (sonido muestreado) a MIDI (sonido a sintetizar) es realmente difícil realizar una transformación de calidad.

4.3.4. Videos

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241

Los vídeos son el contenido multimedia por excelencia. Excluyendo las adaptaciones de los sonidos, en las que se pueden hacer las mismas consideraciones que en el anterior apartado, nos queda un vector de imágenes por adaptar.

Una primera característica a adaptar en los vídeos son los formatos. En este caso la cosa se complica pues recodificar un vídeo tiene una carga computacional muy alta. Pero por suerte no suele ser habitual pues apenas hay codificadores que soporten los terminales (H263 y MPEG4).

Otras características que también se pueden tocar son las dimensiones y el número de imágenes (*frames*) del video. El objetivo sería adaptar el vídeo al tamaño del terminal y al tamaño máximo de fichero del mismo. Aún así también implicaría recodificación del contenido por lo que sólo tiene sentido en caso de vídeos muy cortos.

5. Páginas web de consulta

- www.gsmworld.com
- http://www.onforum.com/tutorials/gsm/
- http://www.wmlclub.com/
- http://www.cellular.co.za/
- http://www.auladatos.movistar.com/Aula-de-Datos/Tutoriales-y-Documentacion/
- http://www.mobilepositioning.com/
- http://telecom.iespana.es/telecom/telef/gsm-info.htm#fr
- http://www.3g-generation.com/
- http://www.moconews.net/
- www.thefeature.com
- http://www.cellular-news.com/
- http://www.openmobilealliance.org/
- http://www.forum.nokia.com/main.html
- http://www.openwave.com/
- http://www.motocoder.com
- http://www.umts-forum.org
- http://www.3gpp.org/
- http://www.itu.int
- http://www.etsi.org/
- http://java.sun.com/j2me/
- http://developers.sun.com/techtopics/mobility/
- http://www.nttdocomo.com/

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

References

1. Alejandro Jiménez-Rodríguez, Luis Fernando Castillo, Manuel González (2012). Studying the mechanisms of the Somatic Marker Hypothesis in Spiking Neural Networks (SNN). ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2
2. André Santos, Regina Nogueira, Anália Lourenço (2012). Applying a text mining framework to the extraction of numerical parameters from scientific literature in the biotechnology domain. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
3. Baruque, B., Corchado, E., Mata, A., & Corchado, J. M. (2010). A forecasting solution to the oil spill problem based on a hybrid intelligent system. Information Sciences, 180(10), 2029–2043. https://doi.org/10.1016/j.ins.2009.12.032
4. Carlos Carvalhal, Sérgio Deusdado, Leonel Deusdado (2013). Crawling PubMed with web agents for literature search and alerting services. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 1
5. Carolina González, Juan Carlos Burguillo, Martín Llamas, Rosalía Laza (2013). Designing Intelligent Tutoring Systems: A Personalization Strategy using Case-Based Reasoning and Multi-Agent Systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 1
6. Casado-Vara, R., Chamoso, P., De la Prieta, F., Prieto J., & Corchado J.M. (2019). Non-linear adaptive closed-loop control system for improved efficiency in IoT-blockchain management. Information Fusion.
7. Casado-Vara, R., Novais, P., Gil, A. B., Prieto, J., & Corchado, J. M. (2019). Distributed continuous-time fault estimation control for multiple devices in IoT networks. IEEE Access.
8. Casado-Vara, R., Vale, Z., Prieto, J., & Corchado, J. (2018). Fault-tolerant temperature control algorithm for IoT networks in smart buildings. Energies, 11(12), 3430.
9. Casado-Vara, R., Prieto-Castrillo, F., & Corchado, J. M. (2018). A game theory approach for cooperative control to improve data quality and false data detection in WSN. International Journal of Robust and Nonlinear Control, 28(16), 5087-5102.
10. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado J.M. (2019). Social computing in currency exchange. Knowledge and Information Systems.
11. Chamoso, P., González-Briones, A., Rivas, A., De La Prieta, F., & Corchado, J. M. (2019). Social computing in currency exchange. Knowledge and Information Systems, 1-21.
12. Chamoso, P., González-Briones, A., Rodríguez, S., & Corchado, J. M. (2018). Tendencies of technologies and platforms in smart cities: A state-of-the-art review. Wireless Communications and Mobile Computing, 2018.
13. Chamoso, P., Raveane, W., Parra, V., & González, A. (2014). Uavs Applied to the Counting and Monitoring Of Animals. In Advances in Intelligent Systems and Computing (Vol. 291, pp. 71–80). https://doi.org/10.1007/978-3-319-07596-9_8
14. Chamoso, P., Rodríguez, S., de la Prieta, F., & Bajo, J. (2018). Classification of retinal vessels using a collaborative agent-based architecture. AI Communications, (Preprint), 1-18.
15. Corchado, J. A., Aiken, J., Corchado, E. S., Lefevre, N., & Smyth, T. (2004). Quantifying the Ocean's CO2 budget with a CoHeL-IBR system. In Advances in Case-Based Reasoning, Proceedings (Vol. 3155, pp. 533–546).
16. Corchado, J. M., & Aiken, J. (2002). Hybrid artificial intelligence methods in oceanographic forecast models. Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, 32(4), 307–313. https://doi.org/10.1109/tsmcc.2002.806072
17. Corchado, J. M., Borrajo, M. L., Pellicer, M. A., & Yáñez, J. C. (2004). Neuro-symbolic System for Business Internal Control. In Industrial Conference on Data Mining (pp. 1–10). https://doi.org/10.1007/978-3-540-30185-1_1
18. Corchado, J. M., Corchado, E. S., Aiken, J., Fyfe, C., Fernandez, F., & Gonzalez, M. (2003). Maximum likelihood hebbian learning based retrieval method for CBR systems. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 2689, pp. 107–121). https://doi.org/10.1007/3-540-45006-8_11
19. Corchado, J. M., Pavón, J., Corchado, E. S., & Castillo, L. F. (2004). Development of CBR-BDI agents: A tourist guide application. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3155, pp. 547–559). https://doi.org/10.1007/978-3-540-28631-8

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

20. Emmanuel Adam, Emmanuelle Grislin-Le Strugeon, René Mandiau (2012). MAS architecture and knowledge model for vehicles data communication. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
21. Fdez-Riverola, F., & Corchado, J. M. (2003). CBR based system for forecasting red tides. Knowledge-Based Systems, 16(5–6 SPEC.), 321–328. https://doi.org/10.1016/S0950-7051(03)00034-0
22. Fernández-Riverola, F., Díaz, F., & Corchado, J. M. (2007). Reducing the memory size of a Fuzzy case-based reasoning system applying rough set techniques. IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 37(1), 138–146. https://doi.org/10.1109/TSMCC.2006.876058
23. Glez-Bedia, M., Corchado, J. M., Corchado, E. S., & Fyfe, C. (2002). Analytical model for constructing deliberative agents. International Journal of Engineering Intelligent Systems for Electrical Engineering and Communications, 10(3).
24. Glez-Peña, D., Díaz, F., Hernández, J. M., Corchado, J. M., & Fdez-Riverola, F. (2009). geneCBR: A translational tool for multiple-microarray analysis and integrative information retrieval for aiding diagnosis in cancer research. BMC Bioinformatics, 10. https://doi.org/10.1186/1471-2105-10-187
25. Gonzalez-Briones, A., Chamoso, P., De La Prieta, F., Demazeau, Y., & Corchado, J. M. (2018). Agreement Technologies for Energy Optimization at Home. Sensors (Basel), 18(5), 1633-1633. doi:10.3390/s18051633
26. González-Briones, A., Chamoso, P., Yoe, H., & Corchado, J. M. (2018). GreenVMAS: virtual organization-based platform for heating greenhouses using waste energy from power plants. Sensors, 18(3), 861.
27. Gonzalez-Briones, A., Prieto, J., De La Prieta, F., Herrera-Viedma, E., & Corchado, J. M. (2018). Energy Optimization Using a Case-Based Reasoning Strategy. Sensors (Basel), 18(3), 865-865. doi:10.3390/s18030865
28. Joana Urbano, Henrique Lopes Cardoso, Ana Paula Rocha, Eugénio Oliveira (2012). Trust and Normative Control in Multi-Agent Systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
29. Laza, R., Pavn, R., & Corchado, J. M. (2004). A reasoning model for CBR_BDI agents using an adaptable fuzzy inference system. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 3040, pp. 96–106). Springer, Berlin, Heidelberg.
30. Manuel Rodrigues, Sérgio Gonçalves, Florentino Fdez-Riverola (2012). E-learning Platforms and E-learning Students: Building the Bridge to Success. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2
31. Mata, A., & Corchado, J. M. (2009). Forecasting the probability of finding oil slicks using a CBR system. Expert Systems with Applications, 36(4), 8239–8246. https://doi.org/10.1016/j.eswa.2008.10.003
32. Méndez, J. R., Fdez-Riverola, F., Díaz, F., Iglesias, E. L., & Corchado, J. M. (2006). A comparative performance study of feature selection methods for the anti-spam filtering domain. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4065 LNAI, 106–120. Retrieved from https://www.scopus.com/inward/record.uri?eid=2-s2.0-33746435792&partnerID=40&md5=25345ac884f61c182680241828d448c5
33. Casado-Vara, R., & Corchado, J. (2019). Distributed e-health wide-world accounting ledger via blockchain. Journal of Intelligent & Fuzzy Systems, 36(3), 2381-2386.
34. Miki Ueno, Naoki Mori, Keinosuke Matsumoto (2012). Picture information shared conversation agent: Pictgent. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
35. Nicholas Beliz, José Carlos Rangel, Chi Shun Hong (2012). Detecting DoS Attack in Web Services by Using an Adaptive Multiagent Solution. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 2
36. Nuno Trindade, Luis Antunes (2013). An Architecture for Agent's Risk Perception. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 2
37. Pawel Pawlewski, Paulina Golinska, Paul-Eric Dossou (2012). Application potential of Agent Based Simulation and Discrete Event Simulation in Enterprise integration modelling concepts. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1
38. Pérez, A., Chamoso, P., Parra, V., & Sánchez, A. J. (2014). Ground Vehicle Detection Through Aerial Images Taken by a UAV. In Information Fusion (FUSION), 2014 17th International Conference on.

ISBN: 978-84-9012-861-9
Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc

39. Prieto, J., Alonso, A. A., de la Rosa, R., & Carrera, A. (2014). Adaptive Framework for Uncertainty Analysis in Electromagnetic Field Measurements. Radiation Protection Dosimetry, ncu260.
40. Prieto, J., Mazuelas, S., Bahillo, A., Fernandez, P., Lorenzo, R. M., & Abril, E. J. (2012). Adaptive data fusion for wireless localization in harsh environments. IEEE Transactions on Signal Processing, 60(4), 1585–1596.
41. Prieto, J., Mazuelas, S., Bahillo, A., Fernández, P., Lorenzo, R. M., & Abril, E. J. (2013). Accurate and Robust Localization in Harsh Environments Based on V2I Communication. In Vehicular Technologies - Deployment and Applications. INTECH Open Access Publisher.
42. Rodolfo Salazar, José Carlos Rangel, Cristian Pinzón, Abel Rodríguez (2013). Irrigation System through Intelligent Agents Implemented with Arduino Technology. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 2, n. 3
43. Tapia, D. I., & Corchado, J. M. (2009). An ambient intelligence based multi-agent system for alzheimer health care. International Journal of Ambient Computing and Intelligence, v 1, n 1(1), 15–26. https://doi.org/10.4018/jaci.2009010102
44. Vasileios Efthymiou, Maria Koutraki, Grigoris Antoniou (2012). Real-Time Activity Recognition and Assistance in Smart Classrooms. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal (ISSN: 2255-2863), Salamanca, v. 1, n. 1

ISBN: 978-84-9012-861-9

Colección Aquilafuente: AQ-241
Ediciones Universidad de Salamanca - cc by nc dc