UNIVERSITAT POLITÈCNICA DE CATALUNYA
DOCTORADO DE INGENIERIA TELEMÁTICA

"CONTRIBUCIÓN AL DESARROLLO DE UN ENTORNO SEGURO DE M-COMMERCE."

TESIS DOCTORAL

Autor: Diego Arturo Ponce Vásquez
Director: Miguel Soriano Ibáñez

Esta Tesis doctoral se ha realizado gracias a una beca AECI concedida por la Agencia Española de Cooperación Internacional.

Esta tesis doctoral la dedico a mi esposa María, a nuestros hijos: Juan Diego, Jenny, Carla y Juan Salvador. A mi hermano Fabián y a mi padre. A la memoria de mi madre y de mi hermana María Soledad.

Agradecimiento:

A Miguel Soriano Ibáñez, tutor y director de esta tesis doctoral, por su gran calidad como docente y como ser humano. Miguel te mereces lo mejor de la vida.

A Pedro Mur Siles, Javier Fernández Bonache, Mónica Roig Ballesté y Sergi Montardit Sicart, proyectistas con quienes hemos compartido el trabajo y camaradería durante estos años en la línea de investigación de m-Commerce.

Al personal académico y administrativo del departamento de doctorado, maestros y amigos.

A mis compañeros de doctorado y del departamento con quienes he compartido su amistad y estos años de esfuerzo y trabajo.

A la Agencia Española de Cooperación Internacional sin quienes estos estudios no hubieran sido posibles.

A todos los que no menciono, pero que con su consejo, amistad y ayuda han llenado estos años de estudio.

Indice

Capítulo 1 Introducción.

1.1 Introducción. 1
1.2 Comercio electrónico. 2
1.2.1 Introducción. 2
1.3 El m-Commerce. 4
1.4 Motivación, Objetivos y Contribuciones de esta Tesis. 5
1.4.1 La seguridad en WAP. 6
1.4.2 La Usabilidad. 7
1.5 Desarrollo de la tesis. 8
1.6 Publicaciones realizadas en el campo. 9

Capítulo 2

2.1 Introducción. 11
2.2 Seguridad en WAP. 12
2.2.1 Modelo de WAP. 12
2.2.2. La Capa de seguridad WTLS 17
2.2.3. Principio de seguridad extremo a extremo. 18
2.3 Soluciones existentes. 19
2.4. Propuesta. 21
2.4.1. WAE-Sec 23
2.5 Conclusiones. 27

Capítulo 3

3.1 Introducción. 29
3.2 Implicaciones de WAE-Sec. 29
3.3 Desarrollo de la capa WAE-Sec. 30
3.3.1 Desarrollo del Cliente. 30
3.3.1.1. Diseño del componente TLS de la capa WAE en el cliente WAP. 31
3.4 Implantación de la capa WAE-Sec. 35
3.4.1. Implantación del cliente WAP. 37
3.4.2 Implantación de la pasarela. 38
3.5 Resultados. 40
3.5.1 El cliente WAP. 40
3.5.2 Resultados en la pasarela. 43
3.6 Comparación de las cargas. 51

Capítulo 4

4.1. Introducción. 54
4.2. Diseño del intermediario. 55
4.3 Sistema de Búsqueda. 59
4.3.1 Funciones del Sistema de búsqueda del Intermediario. 60
4.3.2 Requisitos del sistema de búsqueda 61
4.3.3. Diseño del Sistema Multiagente de Búsqueda 61
4.3.3.1. Componentes funcionales del sistema de búsqueda 62
4.3.4 Seguridad del Sistema multiagente de búsqueda. 63
4.3.4.1 Seguridad de los agentes móviles. 64
4.4 Sistema de Almacenamiento Intermedio. 65
4.5 Sistema de personalización del servicio. 66

4.6 Certificados (OCSP). 67
4.7 Conclusiones. 67

Capítulo 5

5.1 Introducción. 69
5.2 Consideraciones de diseño del intermediario. 69
5.2.1 Esquema del modelo simulado. 70
5.3 Funcionamiento del modelo. 74
5.3.1 Diagrama de Bloques. 75
5.4 Escenarios de prueba. Resultados. 75
5.5 Conclusiones. 77

Capítulo 6

6.1 Introducción. 78
6.2 Implementación del Sistema de búsqueda multiagente. 79
6.2.1 Servidor de Aglets Tahiti. 80
6.2.2 Arquitectura de sistema multiagente. 81
6.2.2.1 Servidor de Aglets Tahiti. 81
6.2.3 Agentes Implantados. 82
6.2.3.1. Agente Listener. 83
6.2.3.2. Agente Principal. 83
6.2.3.3. Agente Base de Datos. 84
6.2.3.4 Agente Cache. 84
6.2.3.5 Agentes Móviles. 84
6.2.4 Implementación y aspectos de seguridad en el sistema de búsqueda multiagente. 87
6.2.5 Funcionamiento del Sistema de búsqueda multiagente. 87
6.3 Sistema de cache y proxy. 89
6.3.1 Implantación. 91
6.4 Sistema de Personalización del servicio. 92
6.5 Repositorio de Certificados y OCSP. 98
6.6 Resultados. 98
6.6.1 Sistema de búsqueda multiagente. 98
6.6.2 Resultados del sistema de cache y proxy. 100
6.6.3 Resultados del sistema de personalización del servicio. 100

Capítulo 7 Conclusiones y líneas futuras de investigación.

7.1 Conclusiones. 103

Referencias. 107

Resumen.

La exitosa implantación de la telefonía móvil a escala mundial presenta una importante oportunidad para la expansión del comercio electrónico sobre entornos inalámbricos. El comercio electrónico para móviles, m-commerce, implica tres aspectos básicos: 1) La negociación y el servicio en la vecindad del cliente, 2) información oportuna y geo-referenciada mientras el usuario está en movimiento, 3) la posibilidad para completar una transacción en cualquier sitio y momento. El usuario debe tener las facilidades siguientes: la negociación y la entrega inmediata, métodos rápidos de micro y macro-pago, y facilidad de uso en el ambiente de móvil. Una de las novedades del comercio móvil es la posibilidad de atraer a clientes en el vecindario hacia un centro de venta y/o servicios proporcionándoles la información apropiada.

Existen, sin embargo, una serie de factores que dificultan la implantación y el desarrollo del comercio móvil respecto al comercio electrónico. Esos inconvenientes se relacionan con las características del ambiente inalámbrico: normalmente menor ancho de banda, latencia más baja, menos estabilidad de conexiones, la disponibilidad menos previsible. Y las limitaciones en los equipos móviles: la unidad de procesamiento central menos potente, menor memoria, limitaciones en el consumo de potencia, formato reducido de pantalla, y otras más.

La seguridad extremo a extremo entre el servidor de Internet y la terminal móvil es también indispensable para aplicaciones de comercio electrónico. La especificación de la capa de seguridad WTLS (Wíreless Transport Layer Security de WAP) no proporciona este nivel de seguridad. El uso de WTLS y la seguridad a nivel de la capa de transporte TLS (Transport Layer Security) permite la privacidad en los canales inalámbricos e Internet, pero la seguridad alcanzada no es suficiente para aplicaciones de comercio electrónico; por ello se precisan mecanismos de seguridad extremo a extremo. Los autores proponen la implementación de una capa de seguridad nueva dentro de la capa WAE (Wíreless Application Environment), denominada por los autores WAE-SEC.

Esto hace posible la seguridad extremo a extremo, compatibilidad con TLS, transparencia ante el usuario, y se evita la traducción y descompresión en la pasarela de WAP. Se propone una alternativa a la arquitectura de la seguridad de WAP para resolver el problema mencionado.

Varios estudios sobre usabilidad de los dispositivos con capacidades de WAP indican que los usuarios se desconectaron debido a tiempos de respuesta lentos y la falta de comodidad en el uso (interfaces no agradables, servicios costosos, ...). Los estudios concordaron que las velocidades más rápidas y el uso extendido de equipos móviles de datos promueven el comercio móvil, y que el número de usuarios familiarizados con equipos móviles sube constantemente, particularmente entre usuarios de WAP, de modo que estos usuarios comienzan a ver sus teléfonos móviles como algo más que meros teléfonos. La movilidad del usuario contribuye a hacer las redes inalámbricas más complejas, y constituyen los nuevos paradigmas en el intercambio de información.

Las posibilidades que se abren para la Internet inalámbrica constituyen una oportunidad importante para el comercio electrónico. El futuro e impacto social de las tecnologías utilizadas, WAP, GPRS, UMTS, ... son cuestionadas habitualmente. En esta tesis se analiza la posibilidad de realizar operaciones de m-commerce sobre WAP. Por ello, se presentan mecanismos de seguridad de WTLS. Y se analiza la seguridad extremo a extremo de este protocolo.

En redes inalámbricas, existe la necesidad de acelerar la respuesta al usuario, facilitarle el uso en ambientes ruidosos, con desconexión no previsible, sobre dispositivos con formato

limitado. Para paliar los problemas relacionados a la entrega de información, facilitar el uso y personalizar el servicio con el cliente, se presenta una propuesta basada en un sistema intermediario. El entorno inalámbrico es un entorno limitado, un intermediario basado en la teoría de las limitaciones, la tecnología de agentes inteligentes y móviles, los sistemas de caches y proxies, y la personalización del servicio, utilizando técnicas de CRM (Customer Relationship Management) tal como se describe en este trabajo, puede contribuir a una mejora estratégica del rendimiento y desempeño global del sistema.

El trabajo presentado en esta tesis doctoral aborda estas dos problemáticas. Por una parte la seguridad extremo a extremo conseguida mediante la introducción de una nueva capa de seguridad, ubicada a nivel de aplicación, compatible con TLS. Por otra parte, la introducción de un intermediario que reduzca el tiempo de respuesta entre el cliente y el servidor, facilite el uso mediante sistemas de búsqueda multi-agente y ofrezca personalización de servicios.

Capítulo 1

Introducción.

1.1 Introducción.

El efecto social de las redes y servicios telemáticos es difícil de predecir. El aumento de ancho de banda disponible será la base de las futuras innovaciones que pueden afectar profundamente la sociedad humana. Por otra parte los dispositivos tienden a adaptarse a las particularidades del usuario brindándole servicios a medida.

Probablemente, la computación móvil e inalámbrica jugará un papel destacado en la industria de Internet en el futuro. Nuevos e interesantes servicios se despliegan al mismo tiempo que Internet entra en nuestras vidas cotidianas. El desarrollo de la infraestructura de la red inalámbrica de banda ancha dará lugar a múltiples y nuevos servicios. El término "computación ubicua" se utiliza a menudo para describir la visión del acceso a Internet en cualquier momento y desde cualquier lugar. Las tecnologías de móviles y las redes inalámbricas jugarán un rol importante en la consecución de esta visión de futuro.

La telefonía móvil ha generado un importante mercado que crece vertiginosamente. Los entornos inalámbricos presentan características propias que los distinguen de los entornos cableados. Como consecuencia surgió la necesidad de un protocolo adecuado para entornos móviles que adapte las soluciones existentes en otros entornos tales como Internet. WAP[1] nace en 1997 con el fin de extender los servicios existentes en Internet al mercado de la telefonía móvil.

Los dispositivos móviles actuales son equipos limitados en capacidad y poseen una interfaz más sencilla de utilizar que los ordenadores personales. Para extender el comercio electrónico desde Internet hacia los entornos móviles, es necesario ampliar la capacidad de estos dispositivos y garantizar la confianza en los mecanismos de negociación y gestión.

El comercio electrónico está cambiando la manera en que los consumidores, comerciantes y empresas realizan sus transacciones. El comercio electrónico permite comprar, invertir, realizar operaciones bancarias, vender, distribuir,... en cualquier lugar donde se pueda disponer de conexión a Internet, y con la interconexión de las redes sin hilos con Internet, desde cualquier lugar y en cualquier momento que se desee.

El amplio potencial de crecimiento del comercio basado en Internet, está siendo frenado entre otros aspectos por las preocupaciones con respecto a la seguridad de la red de comunicaciones. En el comercio electrónico la seguridad es fundamental, se debe garantizar la autenticidad, privacidad, integridad y no repudio en las comunicaciones; de ésto depende su

[1] WAP = Wireless Application Protocol.

aceptación y uso masivo. El riesgo de pérdida de privacidad derivada de posibles problemas en la seguridad de las transacciones y redes de ordenadores corporativas genera reticencia en las empresas para la adopción de técnicas de comercio electrónico. Es pues necesario, poder garantizar la seguridad en las comunicaciones a través de la red.

El uso de teléfonos móviles para el acceso a Internet abre nuevas posibilidades en el comercio electrónico. El m-Commerce[2] involucra tres aspectos básicos; oferta de los negocios y de servicios en un área circundante al usuario; información oportuna, georeferenciada mientras el usuario está en movimiento, y posibilidad de completar la transacción en forma inmediata. Por ello, debe ofrecer al usuario las siguientes prestaciones: negociación y entrega inmediata, métodos de micro y macro pagos, y facilidades de uso en este contexto móvil.

Existen, sin embargo, una serie de factores que dificultan la implantación y desarrollo del m-Commerce frente al e-Commerce. Estos inconvenientes están relacionados con las características del entorno inalámbrico y limitaciones de los teléfonos móviles.

1.2 Comercio electrónico.

1.2.1 Introducción.

El comercio electrónico (e-Commerce) se puede definir, en un sentido amplio, como cualquier forma de transacción financiera o intercambio de información comercial basada en la transmisión de datos sobre redes de comunicación. Sin embargo, dependiendo de cada caso, puede tener diferentes acepciones:

Desde el punto de vista de las comunicaciones es el transporte de información, productos y/o servicios o pagos, mediante canales de comunicación y redes de ordenadores.

Desde la perspectiva de las empresas, es una aplicación de tecnología para la automatización de las transacciones entre organizaciones.

Desde la perspectiva de los servicios, es una herramienta que presenta la oportunidad de rebajar los costes, al tiempo que se aumenta la calidad y la velocidad del servicio prestado.

Finalmente, desde el punto de vista del internauta, es la posibilidad de comprar y vender productos y servicios en Internet, sin tener que desplazarse.

1.2.2 Modelos de comercio.

El comercio electrónico contempla varios aspectos de la negociación y la transacción. El modelo básico consta de un comprador, un vendedor, una entidad financiera y una entidad certificadora. En este modelo, la entrega del producto vendido, si no es tangible, ha sido substituida por un conjunto de información que hace referencia a la descripción del producto y a su entrega, fecha de envío, etc.

El comercio electrónico elimina la comunicación física entre comprador y vendedor, la cual queda substituida por un flujo de información que describe las características del bien vendido. Dicho flujo, se produce también en los dos sentidos, ya que el comprador también facilita información al vendedor.

---

[2] m-Commerce = Comercio electrónico sobre entornos con usuarios móviles.

Básicamente, se pueden establecer tres grandes categorías de comercio electrónico:

B2B[3]*:* Es el comercio electrónico entre empresas para realizar transacciones de negocio. Es el modelo de negocio que más desarrollo ha tenido en los últimos años.

B2C[4]*:* Es el comercio entre empresas y consumidores, generalmente se caracteriza por:

Interactivo. Existe una interactividad continua entre el comprador y el sistema vendedor.
Espontáneo. La comunicación se realiza de forma espontánea a requerimiento del comprador. El vendedor juega un papel pasivo.
Público. Se puede decir que es público, por lo que posee un altísimo potencial.
Global. Internet es implícitamente un canal de distribución que cualquier organización puede utilizar como propio. Es un mercado al que pueden acceder todos los usuarios de la Red.

C2C[5]*:* La negociación se desarrolla entre personas con intereses similares, indistintamente de la parte compradora y vendedora. La comunicación se realiza en forma espontánea y los participantes pueden asumir roles de comprador, vendedor o ambos (intercambio). Requiere sistemas de intermediario para garantizar la confianza entre usuarios.

Destacan las aplicaciones de banca en casa, mediante las cuales el consumidor controla desde su PC el desarrollo de sus operaciones bancarias (consulta de saldo, órdenes de transferencia, compraventa de valores) y las de compra minorista, en la cual el consumidor, puede antes de comprar, utilizar la Red para investigar tipos, calidades y precios. La obtención de información como complemento previo a una compra es una de las características más importantes y diferencial de este tipo de comercio.

Analistas económicos señalan que el éxito del comercio electrónico será proporcional al valor que agregue a la cadena de suministro del producto. La utilización de las nuevas tecnologías facilita al proveedor brindar el servicio de postventa y obtener una realimentación útil para el seguimiento del producto a partir de información suministrada por el cliente. Actualmente existen alternativas tecnológicas para los procesos mercantiles en aspectos tan importantes como procesos publicitarios, estudio y segmentación de mercado, negociación, mecanismos de pago, servicio al cliente, ... El éxito de su implantación masiva, sin embargo, debe aún superar barreras culturales, mejorar la facilidad de uso y garantizar la confianza de los usuarios. Una de las novedades del comercio electrónico es la posibilidad de atraer a los clientes que se encuentran en la vecindad de un centro de negocios y/o servicios aportándoles la información adecuada.

La cadena de valor clásica puede alterarse si, como es posible, cada uno de los eslabones de la misma, ofrece sus productos en este nuevo mercado, tal y como se expresa en la figura 1.1

[3] B2B = Business to Business.
[4] B2C = Business to Consumer.
[5] C2C = Consumer to Consumer.

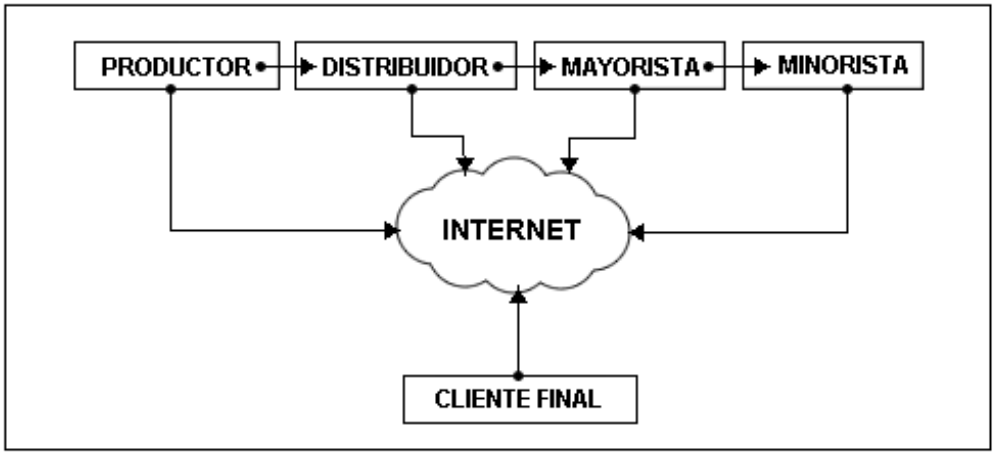


Fig. 1.1 La nueva cadena de valor.

En e-Commerce desaparecen los intermediarios clásicos, ya que la compra es directa. Sin embargo, aparecen otros nuevos componentes clasificados de la siguiente forma:

- Tecnología.
- Información.
- Acceso a la Red.
- Logística y distribución.
- Medios de pago.
- Seguridad, certificación y protección de la propiedad intelectual.
- Gestión del e-Commerce: Afecta a la seguridad que precisa el comprador, las herramientas para buscar mejores ofertas, agentes seguros,...

La seguridad en estos entornos se garantiza habitualmente mediante las siguientes técnicas y protocolos:

- Transporte: con los protocolos TLS[6], WTLS[7]
- Contenidos: protección de la propiedad intelectual (Watermarking, Fingerprinting)
- Acceso: Firewalls, SSH.
- Autoría: firma digital.

1.3 El m-Commerce.

El m-Commerce involucra tres aspectos básicos; i) oferta de los negocios y de servicios en un área circundante al usuario, ii) información oportuna georeferenciada mientras el usuario está en movimiento, y iii) posibilidad de completar la transacción en forma inmediata. Por ello, debe ofrecer al usuario las siguientes prestaciones: a) negociación y entrega inmediata, b) métodos de micro y macro pagos, y c) facilidades de uso en este contexto móvil.

Existen, sin embargo, una serie de factores que dificultan la implantación y desarrollo del m-Commerce frente al e-Commerce. Estos inconvenientes están relacionados con las características del entorno inalámbrico (habitualmente menor ancho de banda, mayor latencia, conexiones menos estables y disponibilidad menos predecible) y limitaciones de los teléfonos

[6] TLS = Transport Layer Security.

[7] WTLS = Wireless Transport Layer Security de WAP.

móviles (procesadores menos potentes, menor memoria, limitaciones en el consumo de potencia, dimensiones de las pantallas, ...).

En el entorno típico de WAP, normalmente el equipo móvil se conecta a Internet a través de una pasarela (WAP proxy) que realiza la traducción entre los protocolos de Internet y WAP.

En el marco del comercio electrónico, existen propuestas interesantes que integran las tecnologías emergentes en una infraestructura de tal naturaleza que se redefine la manera de hacer negocios y acceder a la información en línea. Se pretende implantar una verdadera infraestructura conceptualmente superior a lo que actualmente disponemos en la Web, conocida como "The Grid" [WEST 00].

El mercado de las comunicaciones a móviles puede utilizarse como la extensión del B2B (business to business) para entornos corporativos móviles con clientes internos. En el entorno del consumidor final la situación es del B2C (business to consumer), el acceso libre y sin limitaciones a los contenidos existentes en el Internet. Una manera de comunicar las plataformas de tecnología B2B con las B2C puede ser mediante la utilización de intermediarios que gestionen y faciliten la búsqueda de información al usuario, los intermediarios posibilitan el modelo C2C creando ciberespacios de negocios. En todos los casos, la negociación siempre requerirá protección de los datos.

En el m-Commerce tanto el proveedor como el consumidor se conectan indirectamente a través de entidades de software e Internet, de modo que se deben establecer relaciones de confianza entre las partes y garantizar autentificación, confidencialidad e integridad [THAN 00].

1.4 Motivación, Objetivos y Contribuciones de esta Tesis.

La problemática de los entornos basados en el protocolo WAP, se puede agrupar en dos aspectos fundamentales; el primero la falta de seguridad extremo a extremo. El segundo; la necesidad de gestión rápida de contenidos hacia equipos móviles.

a) Esta tesis propone nuevos mecanismos de seguridad extremo a extremo para realizar las transacciones de comercio electrónico seguro entre los dispositivos móviles y los proveedores de información en Internet.

b) Se propone la utilización de un intermediario de información que facilite el uso y mejore los tiempos entrega de información al cliente móvil. El intermediario se basa en la tecnología de agentes inteligentes con Java para realizar las búsquedas de información, las bases de datos distribuidas en sistemas de caches y proxies para reutilizar los contenidos, y la personalización de los servicios y la relación con el cliente con tecnología CRM[8]. Se analizan los aspectos de seguridad en cada uno de los componentes que intervienen en esta plataforma en el almacenamiento, en la transmisión y la gestión de la información.

Por lo tanto, el objetivo global que se pretende abordar en esta tesis es:

> Proponer un entorno seguro y ágil de comercio electrónico para dispositivos móviles con capacidades de WAP.

Los objetivos específicos son:

[8] CRM = Customer Relationship Management.

- Analizar los mecanismos de seguridad en los entornos Internet y WAP.
- Proponer mejoras de los protocolos de seguridad en WAP.
- Gestionar automáticamente las necesidades de los usuarios móviles y personalizar el servicio.
- Extender las capacidades de los dispositivos móviles, mediante el uso de un intermediario inteligente.
- Aprovechar los estándares y tecnologías existentes para el diseño de un intermediario.

1.4.1 La seguridad en WAP.

En el e-Commerce es necesario crear un ambiente de confianza para todos los participantes. Para conseguirlo, se debe disponer de mecanismos y sistemas de protección contra la piratería, falsificación y delincuencia informática a través de la introducción coordinada de técnicas criptográficas (algoritmos y protocolos), técnicas de control de acceso (firewalls, routers) y de protección de derechos de autor (fingerprinting y watermarking). El comercio electrónico se desarrolla dentro de un entorno de comunicaciones en un ambiente:

- Hostil.
- Vulnerable.
- Con desconfianza mutua entre los comunicantes.

El Comercio electrónico necesita mecanismos eficaces para garantizar la privacidad y la seguridad de las redes abiertas. Estos mecanismos deben proporcionar confidencialidad, autentificación e integridad, y no repudio.

WAP constituye una alternativa a TCP/IP en entornos inalámbricos. El dispositivo móvil se conecta a Internet a través de una pasarela WAP que realiza la traducción usando esta arquitectura.

Los canales seguros entre el cliente móvil y la pasarela WAP soportan WTLS, mientras que el canal entre la pasarela WAP y el servidor en Internet soporta TLS. La traducción entre WTLS y TLS se ejecuta en la pasarela WAP. Lo que no provee la pasarela WAP es seguridad extremo a extremo, la que se define como un canal seguro de comunicación entre las dos partes sobre una red potencialmente insegura. Este hecho comunmente reconocido [KHAR 99], hace que la seguridad en WAP presente las siguientes debilidades:

- No provee autentificación extremo a extremo entre el servidor en Internet y el usuario móvil, y
- Los datos contenidos en los mensajes están en claro en la pasarela WAP.

Propuesta de Seguridad Extremo a Extremo

Con el fin de resolver el problema antes mencionado, existen las siguientes alternativas:

- Colocar la pasarela en el extremo del servidor Web de la conexión, es decir, dentro de la misma zona de seguridad.
- Introducir una capa de seguridad sobre WAP, es decir, considerar WAP meramente como un medio potencialmente inseguro de comunicación.
- Rediseñar el protocolo WAP para no utilizar la pasarela, utilizando los estándares existentes en Internet.

WAP se diseñó para convertir entre dos juegos diferentes de protocolos, uno para red

cableada y el otro para red inalámbrica. La primera solución implicaría el uso inadecuado de los protocolos en las redes cableadas mientras que la tercera solución impide la optimización del protocolo para entornos inalámbricos. La solución propuesta sigue la segunda estrategia. La consecuencia es que algunos de los beneficios que provee la pasarela WAP se pierden [JUUL 01].

Esta propuesta consiste en una nueva capa de seguridad denominada WAE-SEC, compatible con TLS. La ubicación de esta capa se sitúa a nivel de aplicación. Para mejorar la seguridad la nueva capa debe proveer las siguientes características:

Compatibilidad con TLS: TLS es una propuesta de estándar ampliamente aceptado por la comunidad de Internet. Es deseable que los usuarios móviles tengan compatibilidad con los proveedores de Internet sin la necesidad de cambiar o ampliar el protocolo TLS.

Transparencia al usuario: En términos de usabilidad, existe la necesidad de ocultar la complejidad al usuario final. El agente del usuario, o mini-navegador, debería manejar la seguridad de una manera fácil, amigable al usuario. El agente del usuario no forma parte de la especificación de WAP pero debe cumplir las recomendaciones de WAP [WAPF 00],[WAPF 01]. El agente de usuario en WAE[9] debe implantar autentificación básica tal como se especifica en HTTP versión 1.1 (RFC2068). WSP[10] utiliza el mismo juego de caracteres, cabeceras, y códigos de país que HTTP versión 1.1, de modo que es posible establecer un túnel extremo a extremo compatible para realizar transacciones seguras.

Seguridad extremo a extremo: Es necesario que la pasarela WAP no realice la traducción entre WBXML[11] y WML[12] o HTML. La utilización de WAE-Sec en lugar de WTLS, permite al servidor de Internet realizar la traducción en su extremo de la comunicación. Otra alternativa para recibir directamente los mensajes en código WBXML consiste en proveer en el extremo del servidor de los filtros necesarios para traducir las marcas o *tokens*. WAE-Sec provee seguridad extremo a extremo gracias a su ubicación dentro de la pila del protocolo WAP, evitando la traducción y riesgo potencial en la pasarela.

Traducción y Descompresión en la pasarela: Es necesario evitar la descompresión, descifrado y traducción en la pasarela WAP para no comprometer la seguridad extremo a extremo.

1.4.2 La Usabilidad.

Los estudios realizados sobre entornos móviles con WAP coincidieron en que velocidades más rápidas y el uso extendido de equipos móviles de datos promoverían el comercio móvil. La movilidad del usuario contribuye a hacer las redes inalámbricas más complejas, y constituyen los nuevos paradigmas en el intercambio de información.

El canal inalámbrico entre el usuario y la pasarela WAP constituye el "cuello de botella" en cuanto al ancho de banda, por lo tanto el intermediario deberá ubicarse del lado de Internet pero próximo a la pasarela WAP. Para gestionar los contenidos existen esquemas que han tenido éxito en el manejo y replicación de contenidos en sistemas distribuidos Web mediante el uso de caches y proxies [LOON 97]. Esquemas similares pueden ser útiles para gestionar contenidos destinados a terminales móviles.

---

[9] WAE = Wireless Application Environment de WAP.
[10] WSP = Wireless Session protocol de WAP.
[11] WBXML = Wireless Binary Mark-up Language de WAP.
[12] WML = Wireless Mark-up Language de WAP.

La teoría de las limitaciones TOC[13] [GOLD 92] sostiene que un cambio es la mayor parte de las variables en un ambiente complejo tiene (en último término) sólo un impacto pequeño en el desempeño global del sistema. Existen muy pocas variables, en las que un cambio provoca una mejora significativa del rendimiento global del sistema, a estas variables las denomina limitaciones. En las redes inalámbricas, existe la necesidad de acelerar la respuesta al usuario, facilitarle el uso, los ambientes inalámbricos suelen ser ruidosos, con desconexión no previsible y el formato limitado de los dispositivos…

Para paliar los problemas relacionados a la entrega de información, la facilidad de uso y la gestión de la relación con el cliente, se aborda el diseño e implementación de un intermediario inteligente, cuya función básica es gestionar y facilitar el intercambio de información entre cliente e Internet. El intermediario realiza búsquedas de información utilizando agentes inteligentes, utiliza almacenamiento intermedio de contenidos, y gestiona la relación con el cliente utilizando herramientas CRM. Una vez procesada la información se la prepara en el formato de entrega adecuado para el cliente.

A grandes rasgos podríamos considerar dos modos de funcionamiento. En primer lugar, cuando un usuario solicita una determinada información, se realiza un procesado de la solicitud que comienza con la búsqueda de la información que luego será clasificada, organizada, filtrada y entregada al terminal WAP a través del intermediario, tomando las medidas de seguridad necesarias. En segundo lugar, la gestión y entrega automática de contenidos utiliza la información personalizada del perfil del usuario.

1.5 Desarrollo de la tesis.

La estructura de la memoria que es la siguiente:

Capítulo 2. Seguridad extremo a extremo para el comercio electrónico móvil. Describe en detalle la problemática de seguridad y la contribución del mecanismo de seguridad compatible con TLS en a capa de aplicación WAE del WAP.

Capítulo 3. Implantación de la seguridad extremo a extremo con WAE-Sec. Detalla la implantación realizada así como los resultados obtenidos.

Capítulo 4. Usabilidad en WAP. Describe la problemática de usabilidad en el entorno inalámbrico, la necesidad de mecanismos complementarios para gestionar la información y facilitar el uso desde entornos inalámbricos con WAP.

Capítulo 5. Evaluación de las prestaciones del sistema global. Los mecanismos que contribuyen a solucionar las deficiencias en cuanto a seguridad, usabilidad, y tiempo de respuesta son modelados y se realizan cálculos del tiempo de respuesta mediante simulación del entorno.

Capítulo 6. Mecanismos complementarios para la entrega de información: El Intermediario. El diseño de un intermediario con los mecanismos de búsqueda basada en agentes inteligentes, sistemas de cache y proxies y la personalización del servicio que contribuyen a mejorar el tiempo de respuesta, facilitar el uso y asistir al usuario de aparatos móviles en la gestión personalizada de la información de forma adecuada al entorno sin hilos.

[13] TOC = Theory of Constraints.

Capítulo 7. Conclusiones. Se citan los resultados obtenidos, las prestaciones y limitaciones de los mecanismos propuestos y se enumeran las líneas de investigación futura.

Referencias. Se citan las referencias bibliográficas y las publicaciones vinculadas con el tema.

Anexos. Contiene el código y documentación generada como producto de los años de investigación en el tema de esta tesis.

1.6 Publicaciones realizadas en el campo.

A continuación se citan las publicaciones realizadas durante la fase de investigación del tema:

Seguridad Extremo a extremo:

[PONC 00] D. Ponce, M. Soriano, P. Mur, "A proposal for B2C Electronic Commerce Scheme based on WAP". World Multiconference on Systemics, Cybernetics and Informatics Proceedings Vol. IX, (Jul. 2000), pp. 537-542. ISBN 980-07-6695-2

[SORI 00] M. Soriano, R. Gonzalo, D. Ponce, P. Mur, "Comercio electrónico seguro a través de WAP". Simposio Español de Informática Distribuida. Libro de Actas, (Sep. 2000), pp. 295-303. ISBN 84-8158-163-1

Mediador para entornos móviles:

[PONC 01a] D. Ponce, M. Soriano, "Intermediario Inteligente para m_Commerce", III Jornadas de Ingeniería Telemática Jitel 2001 (Sep. 2001), pp. 175-181. ISBN 84-7653-783-2.

Usabilidad:

[SORI 01] M. Soriano, D. Ponce, "Mediador inteligente para comercio electrónico en entornos móviles". II Jornadas de Ingeniería Informática. UCAB – Venezuela, Ponencia invitada. (Feb. 2001)

[PONC 01b] D. Ponce, M. Soriano, "Secure Intelligent Broker for m-Commerce". International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, SSGRR 2001, (Ago. 2001), ISBN 88-85280-61-7.

[PONC 01c] D. Ponce, M. Soriano, J. Fernandez, "Electronic Commerce Scheme based on WAP". (Sometido al proceso de evaluación en Journal of Electronic Commerce Research).

Síntesis global del sistema:

[PONC 01d] D. Ponce, M. Soriano, "Seguridad Extremo a Extremo para Comercio Electrónico Móvil y Mecanismos Complementarios para la Entrega de Información". I Simposio Español de Negocio Electrónico (SNE'01), (Oct.2001), pp. 231-251.

[SORI 02] M. Soriano, D. Ponce, "Security and Usability Proposal for Mobile Electronic Commerce", Feature Topic of IEEE Communications Magazine on Evolving

Seamless All IP Wireless and Mobile Networks, (aceptado, IEEE).

Caracterización del tráfico WAP con intermediario:

[PONC 02] D. Ponce, M. Soriano, F. Barceló, S. Montardit, “Complementary mechanisms for Reliable an d Secure Mobile Electronic Commerce”, IASTED 02, (2002). (Sometido a Proceso de revisión.)

Capítulo 2

Seguridad extremo a extremo para el comercio electrónico en entornos móviles.

2.1 Introducción.

Internet se extiende por el mundo, prácticamente cualquier persona conoce su existencia y un porcentaje considerable lo utiliza. Se podría decir que se está convirtiendo en una herramienta prácticamente imprescindible para la sociedad y por tanto su expansión, estudio, legislación y complejidad también crecen a pasos agigantados. El comercio por Internet es el que crece a mayor velocidad y son cada vez más empresas las que se están proporcionando estos servicios: es el denominado comercio electrónico. Por otro lado, el uso de las comunicaciones móviles se extiende incluso a mayor velocidad que Internet. Cada vez son más fabricantes y más operadoras telefónicas las que ofrecen servicios de telefonía móvil.

En este entorno, Internet no es una red segura ya que no fue diseñada para serlo. El problema de la seguridad es bastante serio ya que en las transacciones propias del comercio electrónico viajan por la red datos secretos del usuario como pueden ser su número de tarjeta de crédito o sus datos bancarios y por tanto deben ser protegidos. De esta necesidad surgieron los protocolos de seguridad entre ellos TLS como propuesta de estándar para proporcionar confidencialidad, integridad y autenticación en comunicaciones que usan TCP.

La telefonía móvil ofrece una serie de posibilidades que lo convierten en una herramienta fundamental. Ambas tecnologías se funden en una sola y podemos tener Internet en un terminal móvil, combinar la capacidad de Internet en un entorno donde el usuario pueda moverse y disponer de conexión las 24 horas del día, en cualquier lugar [FASB 99]. De esta idea surge WAP, la arquitectura de protocolos que especifica como acceder a Internet desde un terminal móvil. La familia de protocolos TCP/IP presenta una serie dificultades [KARN 99] al momento de trabajar en entornos inalámbricos móviles. Estos factores unidos al ancho de banda limitado por la telefonía móvil condicionan a los fabricantes mundiales a constituir el consorcio WAP Forum para desarrollar una nueva pila de protocolos adecuada a los entornos inalámbricos con usuarios en movimiento.

Aunque WAP fue diseñado para utilizar cualquier tecnología móvil existente, la más utilizada por WAP es GSM. GSM es una tecnología digital de acceso aéreo que incluye mecanismos de cifrado de la comunicación entre el terminal móvil y la estación base (BSC). No va a ser misión de este trabajo describir el funcionamiento de GSM, pero debe señalarse que durante los últimos años han surgido dudas sobre la fortaleza de los algoritmos utilizados [SORI 96], [GONZ 98]. Se considera comúnmente que los mecanismos de cifrado de GSM no son suficientes para garantizar la seguridad de cualquier transacción conducida mediante WAP, debido no sólo a la debilidad de los algoritmos como a la porción de camino protegidas (exclusivamente desde el terminal móvil a la BTS).

WAP se articula como una arquitectura en capas en la que la capa de transporte se denomina WDP (Wireless Datagram Protocol) [WAPF 99],[WAPF 01]. Sobre esa capa de transporte se sitúa una capa opcional de seguridad, denominada WTLS, Wireless Transport Layer Security. Del mismo modo que en el mundo TCP/IP se ha consolidado un estándar de facto, TLS se ha impuesto como capa de seguridad entre los protocolos de aplicación (HTTP, FTP, SMTP, etc.) y la capa de transporte, la especificación WAP ha definido WTLS. Este protocolo se ha diseñado teniendo en cuenta los siguientes criterios:

- WTLS debe soportar datagramas.
- Debe soportar portadoras de ancho de banda reducido y diverso.
- Debe soportar periodos de latencia potencialmente largos.
- La capacidad de memoria y procesamiento de los terminales puede ser pequeña.

En definitiva, TLS y WTLS son protocolos equivalentes (en múltiples partes de la especificación de WTLS se copia literalmente la de TLS), siendo patente que la intención de los autores del protocolo fue coger TLS y añadir soporte a datagramas, optimizar el tamaño de los paquetes transmitidos y seleccionar algoritmos rápidos entre los permitidos.

2.2 Seguridad en WAP.

2.2.1 Modelo de WAP.

El modelo de aplicación WAP (Figura 2.1) es bastante similar al WWW, ya que todo el sistema WAP está basado en el anterior. Este parecido permite facilidades tales como un modelo de programación familiar, una arquitectura probada y la habilidad de utilizar herramientas existentes (Servidores Web, Herramientas XML, estándares de Internet,...). También debe indicarse que se ha intentado optimizar el modelo para un entorno inalámbrico.

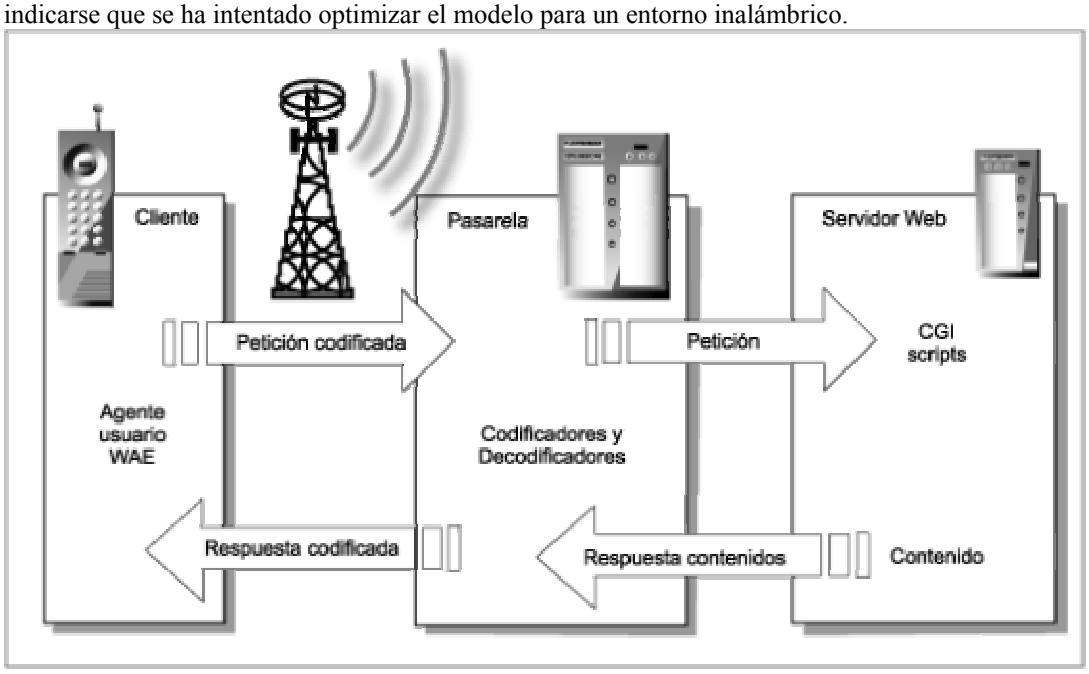


Fig. 2.1 Modelo WAP.

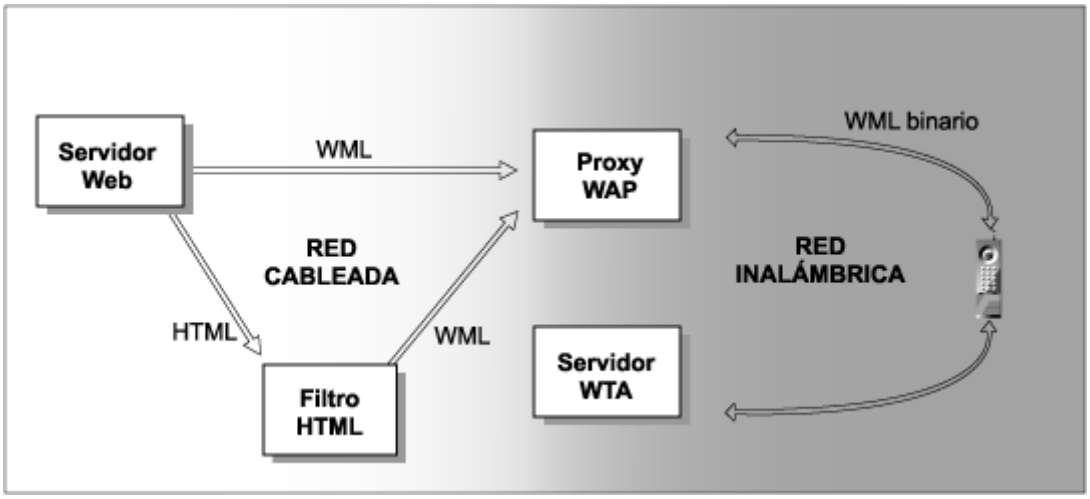


Fig. 2.2 Modelo de programación de la red de WAP.

Como se puede desprender de la Figura 2.2, el modelo opera de la siguiente manera:

1. El usuario teclea la URL en su teléfono móvil y pulsa ver.
2. El agente usuario envía la petición URL a la pasarela WAP mediante el protocolo WAP.
3. La pasarela WAP genera una petición convencional HTTP para la URL pedida y la envía al servidor Web.
4. El servidor Web procesa la petición. Si es un fichero estático, toma el fichero y le añade una cabecera HTTP. Si es un CGI (Common Gateway Interface) u otra aplicación *Script*, lanza la aplicación.
5. El servidor Web devuelve la marca WML con la cabecera HTTP añadida, o la salida WML del CGI o *Script*.
6. La pasarela WAP verifica la cabecera HTTP y el contenido WML y la codifica a una forma binaria. Crea la respuesta WAP conteniendo el WML y lo envía al agente usuario.
7. El agente usuario recibe la respuesta WAP y muestra por pantalla el contenido WML o *Script*.

El contenido se transporta usando la torre de protocolos. Además, se dispone de un micro-navegador en el terminal móvil que hace de interfaz con el usuario.

WAP define un conjunto de componentes estándares que permiten la comunicación entre el cliente móvil y los servidores que deben incluir:

- Modelo de nomenclatura: se utilizan los URLs estándar.
- Representación del contenido: contenido consistente con el WWW.
- Formatos de contenido estándar: basados en WWW además de incluir información de calendario, tarjetas electrónicas de negocio, imágenes y lenguaje Script.
- Protocolos estándar: permiten la comunicación entre el navegador del dispositivo inalámbrico y el servidor.

WAP utiliza la tecnología *Proxy* para conectar el dominio inalámbrico al Internet tradicional. Entre el terminal móvil y el servidor Web existe una pasarela. En este nodo se traducen los datagramas del protocolo WAP al protocolo HTTP-TCP/IP. Por tanto, el cliente, desde su terminal con capacidades WAP ve esta pasarela como el extremo de la comunicación. A modo de resumen sus características principales son las siguientes:

- Pasarela de protocolos: traduce las peticiones de la torre WAP (WSP, WTP, WTLS y WDP) al protocolo WWW (HTTP y TCP/IP).

- Codificadores y decodificadores de contenido: traducen el contenido WAP en formatos codificados compactos para reducir la carga sobre la red.

Entorno de programación de WAP.

El cliente WAP se comunica con dos servidores en la red inalámbrica. La pasarela WAP traduce las peticiones WAP en peticiones WWW y también en dirección contraria (respuestas WWW en respuestas WAP).

Si el servidor Web proporciona directamente contenido WAP (WML), la pasarela WAP lo coge directamente del servidor. Sin embargo, si el servidor sólo proporciona contenido WWW (HTML), se utiliza un filtro para traducir contenido WWW en contenido WAP (HTML → WML). Las marcas WML son codificadas a WBXML antes de enviarlas al móvil WAP.

El servidor de Aplicación de Telefonía Inalámbrica, WTA (Wireless Telephony Application) es un ejemplo de servidor que responde peticiones directamente del cliente WAP sin pasar por ningún tipo de intermediarios. Se utiliza fundamentalmente para aplicaciones propias del entorno inalámbrico.

Modelo de referencia WAP

El modelo de interconexión de capas en WAP es el ilustrado en la Figura 2.3. Los protocolos WAP y sus funciones están construidos sobre un modelo de capas basándose en el modelo de referencia OSI de la ISO. Las entidades de gestión de las capas del protocolo manejan la inicialización, la configuración y las condiciones de error.

La capa de Aplicación WAE

La capa de aplicación (Wireless Application Environment) es la capa de propósito general basada en una combinación de World Wide Web (WWW) y las tecnologías de telefonía móvil. Esta capa se explicará más a fondo debido a la importancia que tiene dentro del trabajo realizado. Su principal objetivo es establecer un entorno de interoperabilidad que permitirá a los usuarios y los proveedores de contenidos construir aplicaciones y servicios que puedan alcanzar una gran variedad de plataformas inalámbricas de manera eficiente y útil. WAE incluye un mini-navegador con las siguientes funcionalidades:

- Wireless Mark-up Language (WML) – un lenguaje liviano, similar al HTML, pero optimizado para uso en terminales móviles. Wireless Binary Mark-up Language, (WBXML), es la versión codificada que se entrega a los dispositivos móviles para reducir el volumen de tráfico hacia el móvil.

- WMLScript – un lenguaje Script de baja carga, similar a JavaScript;

- Wireless Telephony Application (WTA, WTAI) – servicios de telefonía e interfaces de programación.

- Formatos de contenidos – un conjunto de formatos de datos bien definidos, incluyendo imágenes, agenda, e información de calendario.

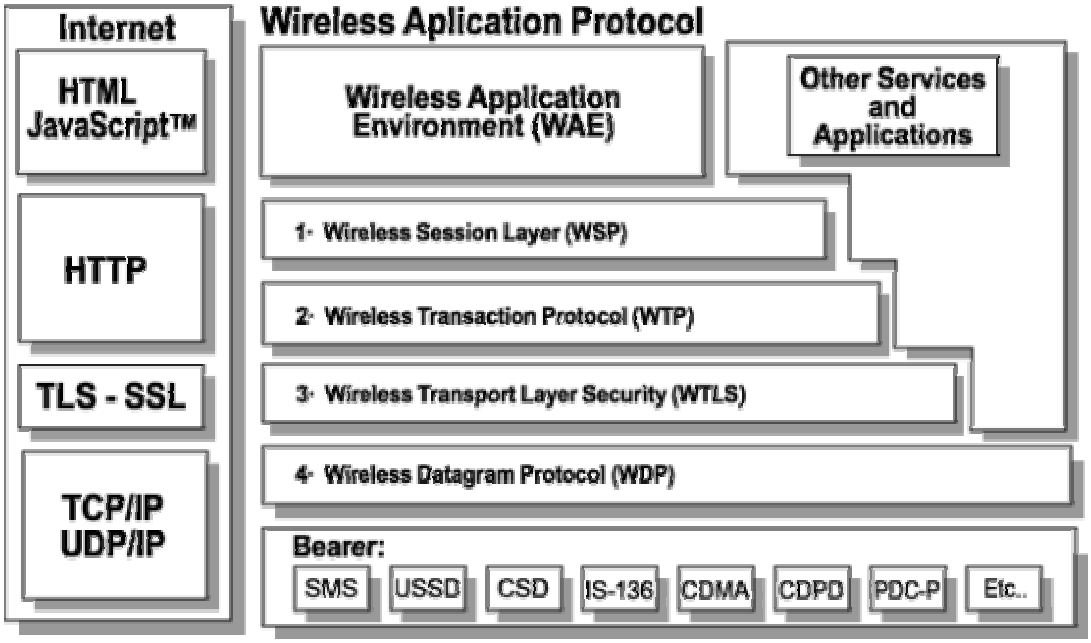


Fig. 2.3 Pilas de protocolos TCP/IP y WAP.

Modelo WAE

La arquitectura WAE permite procesar adecuadamente los contenidos y los servicios provenientes de los servidores Web actuales. Dichos contenidos están localizables utilizando los URLs estándar.

WAE mejora algunos de los estándares WWW con el propósito de adecuarlos a las características de los dispositivos y redes, se considera dentro del modelo la existencia de una pasarela encargada de codificar y decodificar los datos con el fin de minimizar tanto la carga que viaja por el aire como el coste computacional requerido por parte del cliente para procesar los datos. Los principales elementos del modelo WAE son los siguientes:

- Agentes usuario WAE. Es el software del dispositivo del cliente que proporciona funcionalidad específica al usuario final. Los agentes usuario tales como navegadores están integrados en la arquitectura WAP. WAE incluye los agentes usuario para los dos principales tipos de contenidos estándar: WML y WMLScript.
- Generadores de contenido. Aplicaciones (o servicios) en servidores origen (CGI, Scripts, ASPs, ...) que producen contenido a raíz de la petición de los terminales móviles.
- Codificación de contenidos estándar. Permiten a un agente usuario navegar por el contenido Web. Incluye compresión de WML y WMLScript, formatos estándar de imagen, y formatos de calendario y tarjetas de negocio.
- Aplicación de telefonía inalámbrica. Colección de extensiones específicas de telefonía para las llamadas telefónicas.

Normalmente, es el agente usuario quien inicia la petición de contenido. Sin embargo, WTA incluye mecanismos que permiten a los servidores repartir contenido a los clientes sin la necesidad de la petición previa, el Push. Este servicio puede parecer un tanto inapropiado porque lo más usual es que sea el cliente (el que a fin de cuentas paga por la conexión) quien

pida una página WML cuando quiera. Sin embargo como se verá a lo largo de la memoria, puede haber ocasiones en las que este tipo de servicio puede ser muy útil.

Nomenclatura URL

La arquitectura WAE utiliza las URLs de WWW asumiendo:

- La existencia de una arquitectura generalizada para describir el comportamiento de la pasarela para los diferentes tipos de URLs.

- Soporte para la conexión de al menos una pasarela WAP.

La mayoría de conexiones entre el navegador y la pasarela utilizan WSP, independientemente del protocolo del servidor destino. Sin embargo, la URL, siempre especifica el protocolo que se usa en el servidor destino sin tener en cuenta el protocolo del navegador para conectarse a la pasarela.

Componentes del WAE

Como se puede ver en la Figura 2.4, la capa WAE se divide en 2 capas lógicas:

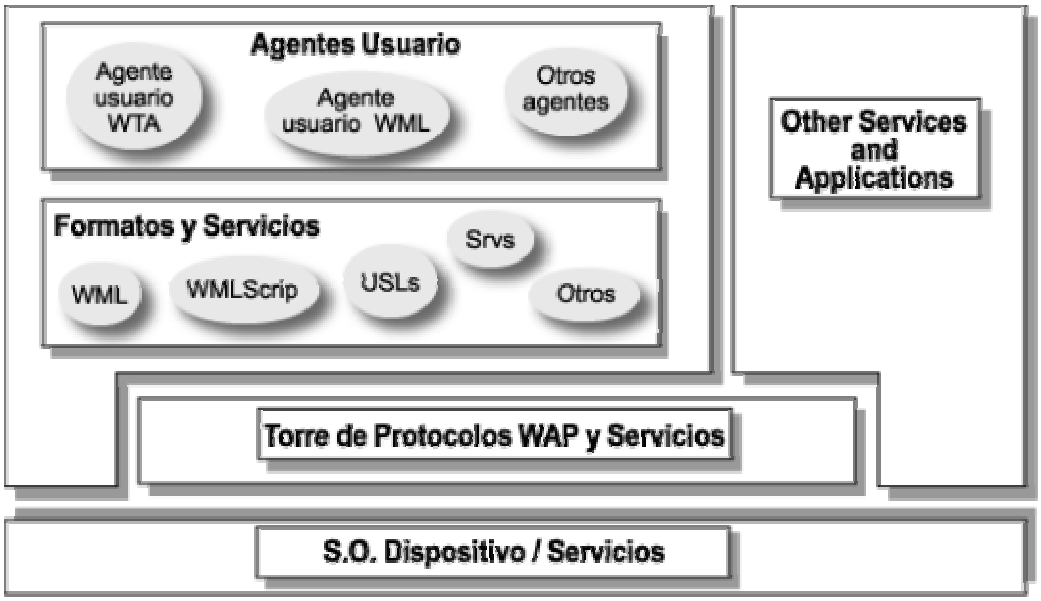


Figura 2.4. Componentes del cliente WAE

- Los agentes usuario, que incluyen los navegadores, agendas, etc.

- Servicios y formatos, que incluyen el conjunto de elementos y formatos accesibles por los agentes del usuario como son el WML, WMLScript, formatos de imagen, vCard, etc.

WAE separa las dos capas y asume un entorno con múltiples agentes usuario. También los formatos son múltiples, y por tanto, las combinaciones entre ellas son múltiples y serán definidas por el diseño de los desarrolladores.

WML

WML es un lenguaje basado en XML, y es la base de los contenidos WAP. WML es un lenguaje de documentos basado en marcas (*tais)*, como HTML y HDML de WWW.

WML se especifica como un documento XML y está optimizado para presentaciones específicas e interacción con el usuario en dispositivos de capacidad limitada como teléfonos móviles y otros terminales móviles de tecnología inalámbrica.

WML y el entorno que soporta fueron diseñados con ciertas limitaciones propias de dispositivos de banda estrecha; (pequeñas pantallas, facilidades limitadas sobre la capacidad del usuario de introducir datos, estabilidad de las conexiones, recursos limitados de memoria y de capacidad de computación). Y por lo tanto, una de las principales dificultades es la de distribuir correctamente la presentación de los contenidos sobre la pantalla del dispositivo WAP.

Una pantalla de WAP se denomina *card*, un conjunto de cards forma un *deck*. El agente de usuario debe decidir cual es la presentación adecuada para una *card* dependiendo de las prestaciones del dispositivo en el que actúa. Por ejemplo algunos dispositivos con pantallas de mayor tamaño pueden fusionar varias *cards*, o puede darse el caso contrario en que sea necesario fraccionar la *card* activa en varias pantallas.

2.2.2 La Capa de seguridad WTLS.

La Capa de Seguridad WTLS es un protocolo basado TLS, utilizado en el WWW para la provisión de seguridad en la realización de transferencias de datos. Este protocolo ha sido diseñado para proveer seguridad a nivel de la capa de transporte y optimizado para ser utilizado en canales de comunicación de banda estrecha de WAP.

Para este protocolo se han definido las siguientes características: integridad de los datos, privacidad de los datos, autentificación y no repudio.

Aunque los dos protocolos son bastante parecidos y el WTLS está basado en TLS, existen algunas diferencias entre ambos debido a los entornos diferentes en los que trabajan y la capacidad de cálculo criptográfico de los dispositivos [KHAR 99]. WTLS redefine los códigos de alerta de TLS existentes además de añadir nuevos como pueden ser la notificación de cierre de sesión, no-conexión, identificador de clave desconocido, identificador de clave deshabilitado, sesión no preparada, etc.

En el modelo general el dispositivo móvil se conecta a Internet a través de una pasarela entre protocolos de Internet y WAP. En el modelo se puede observar que la seguridad entre el cliente y pasarela de WAP lo soporta WTLS, mientras el canal entre pasarela de WAP y servidor de Internet lo soporta TLS. La traducción entre WTLS y TLS se ejecuta en la pasarela de WAP. Para garantizar la privacidad y la integridad de datos en pasarela de WAP se requiere lo siguiente:

- Nunca almacenar datos en claro en la memoria secundaria.
- Utilizar cifrado y descifrado rápido con el borrado del contenido de la memoria interna volátil inmediatamente después que la transacción se ha realizado.
- Asegurar físicamente el acceso a la consola de la pasarela de WAP y restringir el acceso solamente para el personal administrador autorizado.
- Aplicar todos los mecanismos de seguridad necesarios para proteger los sistemas de facturación y registro de ubicación en la pasarela de WAP.

Seguridad WAP entre cliente y servidor

La utilización de WTLS para la realización de comunicación segura entre terminales, por ejemplo en el caso de operaciones de comercio electrónico entre terminales móviles es tal y como se muestra en la Figura 2.5.

En la parte de la derecha, la pasarela recoge los mensajes codificados con TLS del servidor Web y los convierte a la capa de seguridad WTLS, las peticiones desde el teléfono hacia el servidor Web, recorren el camino inverso.

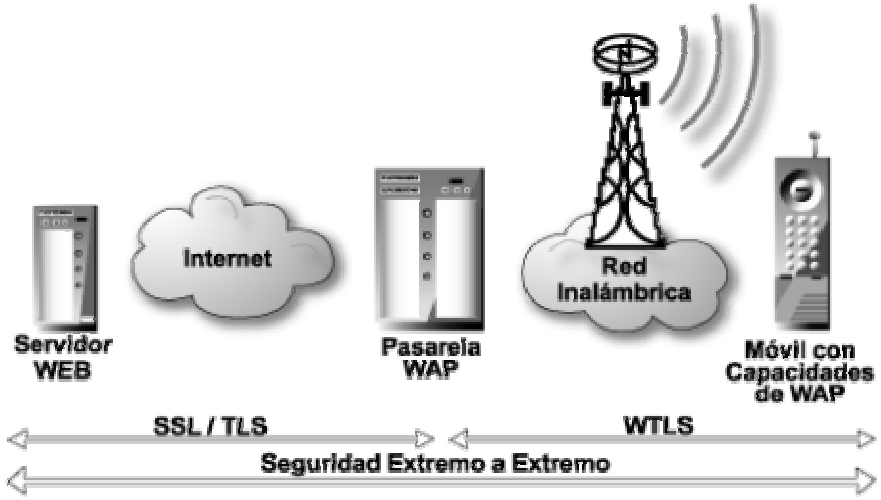


Fig. 2.5 Seguridad en WAP.

Debilidades de WTLS

Las principales debilidades que tiene WTLS son las siguientes:

- No ofrece procedimientos de autentificación extremo a extremo. La autenticidad del mensaje debe ser verificada por la aplicación.
- No se garantiza privacidad en la pasarela WAP, debido a que los datos están en claro y por tanto son potencialmente vulnerables mientras se preparan para ser enviados al nodo destino.

Si la aplicación realizase el cifrado extremo a extremo, obtendría su autentificación, manejaría las claves a su satisfacción, y no se expondrían los datos fuera de la aplicación. También existen diversos ataques que se pueden realizar a los componentes de seguridad de WTLS por la manera como está definido el protocolo. Por ejemplo hay un ataque conocido al MAC como se puede ver en [CHRI 00]. La clave intercambiada, el cifrado y el algoritmo de MAC se negocian independientemente. Esto conlleva una gran flexibilidad que permite conseguir combinaciones sin sentido como podrían ser intercambio de clave nula (NULL) con integridad o protección de confidencialidad. El CBC se calcula para cada registro debido a que el orden de los registros no está garantizado, pero existen los números de secuencia que garantizan el orden de los registros.

En [SAAR 99] se ponen de manifiesto diversos ataques criptográficos puramente matemáticos contra el protocolo WTLS. No es motivo del presente documento explicar las debilidades matemáticas del protocolo pero se enumerarán las principales: IVs predecibles con ataques contra secretos de baja entropía, el XOR MAC y los cifrados de flujo, cifrado con DES de 35 bits, el ataque PKCS #1 (las firmas RSA y el cifrado están diseñados de acuerdo con PKCS #1, versión 1.5), mensajes de alerta no autentificados y ataques a textos probables.

2.2.3 Principio de seguridad extremo a extremo.

El principio de diseño extremo a extremo [SALT 84] guía la ubicación de funciones entre los módulos de un sistema de computación distribuida. El principio, denominado argumento de extremo a extremo en diseño de sistemas, sugiere que las funciones ubicadas en las capas de bajo nivel de un sistema pueden ser redundantes o añaden poco valor frente al coste de proveerlos. El aspecto que nos interesa está relacionado con la transmisión segura de datos

utilizando criptografía. Para que una transmisión de datos entre dos máquinas sea segura; la aplicación de los algoritmos de seguridad se debe hacer en las capas altas de la torre de protocolos, siendo siempre recomendables la negociación y la autentificación entre un extremo y el otro extremo de la comunicación. Este argumento se basa en tres consideraciones:

- Si la transmisión de los datos utiliza cifrado y descifrado, las claves requeridas, deben ser gestionadas de una manera fiable.
- Los datos estarán en claro, por tanto vulnerables, mientras pasan por la pasarela y son transmitidos a la aplicación destino.
- La autenticidad de los mensajes debe ser comprobada por la aplicación.

Si la aplicación utiliza cifrado extremo a extremo y obtiene su autentificación entonces puede manejar a su satisfacción las claves, y los datos nunca son expuestos fuera de la aplicación.

Para satisfacer los requisitos de la aplicación, no es necesario que el subsistema de comunicación provea un cifrado automático para todo el tráfico. Se debe también garantizar que el mal comportamiento del usuario o del programa de aplicación no transmita deliberadamente información que no debería ser expuesta. La única solución que cumple todos estos requisitos consiste en la utilización de un mecanismo de extremo a extremo apropiado para poder tener un acceso seguro a través de Internet.

2.3 Soluciones existentes.

Desde el punto de vista del proveedor de contenidos (o de servicios), básicamente existen tres modelos de prestación de servicios WAP:

a) El proveedor de contenidos integra su servicio en un portal móvil de un operador (e-moción, Conecta o amen@wap, por citar los españoles). En este caso, el propietario de la pasarela, el operador móvil, firma un contrato con el proveedor de contenidos por el que garantiza la seguridad de los datos que transitan por la pasarela (sólo integridad y confidencialidad) y, opcionalmente, estableciendo la comunicación con el proveedor de contenidos mediante una VPN, consigue así garantizar la seguridad de los datos en tránsito. La seguridad del tramo WAP se haya protegido explícitamente con WTLS y la del tramo Internet con TLS.

b) El proveedor de contenidos utiliza la infraestructura de cualquier operador móvil para garantizar el acceso a su servicio o contenido, pero sin haber llegado a ningún acuerdo con el operador (el usuario accede al servicio introduciendo una URL en tras haber accedido a la conectiva WAP del operador). En este caso, no se garantiza ningún servicio de seguridad, puesto que depende de la configuración de la pasarela del operador ofrecer TLS en el tramo Internet (incluso aunque se solicite una URL utilizando el protocolo HTTPS). Si la presencia de WTLS también depende de la configuración de la pasarela, puesto que aunque sea requerida por el usuario, generalmente no será implementada.

c) El proveedor de contenidos decide garantizar la seguridad, ofreciendo los servicios de acceso al usuario. Para ello, el usuario tendrá que crear un nuevo perfil dentro de su teléfono WAP para asegurar la conexión al servicio (número de teléfono, pasarela WAP...) o indicar al proveedor de contenidos cual es su número de teléfono para que éste configure vía OTA el teléfono del usuario. En este caso, el proveedor de contenidos puede adquirir y configurar su propia pasarela WAP garantizando la existencia de WTLS en el tramo WAP de la comunicación. La existencia de TLS no es necesaria puesto que, o bien el servidor de origen se halla dentro de la intranet del proveedor o bien utiliza un servidor WAP en vez de una pasarela (como el servidor WAP de Nokia), integrando los contenidos con la infraestructura WAP.

Mecanismos de one-time password [LAMP 81], que en el mundo Internet se implementan mediante calculadoras lógicas realizadas con JavaScript de cliente, no son posibles con los terminales actuales, más por restricciones del tamaño de los paquetes transmitidos que por la limitación de potencia de proceso o memoria de dichos terminales.

El elemento novedoso que parece aportar la tecnología WAP es la posibilidad de utilizar los elementos de autenticación propios de la red GSM. Sin embargo, esto no es sencillo de conseguir de modo directo, utilizando las funciones de la tarjeta SIM, sino que hay que tomar un rodeo. Una vez autenticado el terminal en la red GSM, el operador telefónico puede traspasar el número de teléfono al servidor Web del proveedor de contenidos en forma de cabecera HTTP. La pasarela WAP de phone.com (UP.Link) lo hace mediante la cabecera X_UP_SUBNO, en tanto que la de Ericsson utiliza MSISDN (si bien este número va cifrado). El WAP Server de Nokia y Kannel envían la cabecera x-network-info.

Es preciso tener mucho cuidado con estas cabeceras puesto que su traspaso indiscriminado proporcionando el número de teléfono del usuario puede plantear ataques a la intimidad de éste. De hecho el operador estadounidense Sprint tuvo problemas, utilizando UP.Link, puesto que proporcionaba en claro dicho número de teléfono. Por tanto, no puede confiarse demasiado en la presencia de esta cabecera, a no ser que, como hace la pasarela de Ericsson, el número de teléfono se traspase cifrado. De este modo, sólo la aplicación del proveedor de contenidos que posea la clave para descifrar este número podrá conocerlo.

WAP 1.2 introduce nuevas posibilidades que aumentan la seguridad proporcionada por la familia de protocolos. Se trata de WIM (Wireless Identity Module) y de una nueva biblioteca de funciones (la sexta) de WMLScript con propósitos criptográficos (Crypto) [PHONE 00].

Aunque las especificaciones WAP 1.2 están aprobadas desde diciembre del año 1999, la infraestructura y terminales disponibles en la actualidad en el mercado soportan exclusivamente las características esenciales de WAP 1.1, como WTLS. En Junio del 2001 se liberó la versión 2.0 de WAP, la que incluye soporte de algoritmos criptográficos en la capa de aplicación WAE.

WIM es una especificación que trata de definir un equivalente en el ámbito de WAP del popular SIM (Subscriber Identity Module), implementado en las tarjetas de nuestros teléfonos GSM y que contenía la identidad del usuario en la red GSM. WIM es una aplicación para tarjetas inteligentes (la propia tarjeta SIM es una tarjeta inteligente) con varios propósitos:

- Almacenar el par de claves del usuario, el certificado que avala dichas claves y cualquier certificado raíz.
- Almacenar las claves simétricas de sesión.
- Efectuar las operaciones criptográficas necesarias para ejecutar los procedimientos de la capa de seguridad (firmado, generación de claves)...
- Al tratarse WIM de una aplicación para tarjetas inteligentes, puede implementarse en una tarjeta aparte (válida para teléfonos dual slot, dual SIM o que se encuentren conectadas a un lector de tarjetas mediante infrarrojos o Bluetooth) o almacenada en una tarjeta multiaplicación que contenga otras aplicaciones como la SIM de GSM (este modelo pone el control de las claves del usuario en manos de su operador móvil, puesto que es éste el que le provee de tarjeta SIM). El acceso a las funciones de WIM se protege también mediante un PIN.

WMLScript Crypto API no es más que una nueva biblioteca estándar de WMLScript. De momento, sólo contiene una función (Crypto.signText), similar a la ya disponible en JavaScript desde la especificación 1.3 [NETS 98]. Su finalidad, como puede deducirse fácilmente, es generar una firma digital de un texto que es enviado al terminal WAP dentro de una deck.

Utilizando esta función para generar una firma digital sobre un contrato digital o un resguardo de una transacción, se consigue la irrenunciabilidad con prueba de origen.

Finalmente, en la actualidad, existen varias especificaciones en discusión en el WAP Forum para solucionar este problema. La más significativa es la que permitirá la "tunelización" de conexiones a través de una pasarela, de forma que una conexión WAP sea atendida por la pasarela del operador y traspasada a la del proveedor de servicios, estableciéndose WTLS entre el terminal WAP del usuario y la pasarela WAP del proveedor de servicios. Esta especificación

se denomina Wireless Port Proxy. Entre los avances que se están estudiando en materia de seguridad, algunos ya han sido citados:

- La posibilidad de traspasar conexiones WAP entre pasarelas de forma que el *white spot* se produzca en la infraestructura del proveedor de contenidos (Wireless Port Proxy).
- La posibilidad de almacenar los certificados de cliente en un servicio de directorio (como LDAP) y referenciarlos mediante una URL.

Otros de los planteados son los siguientes:

- En futuras versiones de WMLScript Crypto API, se añadirán funciones de cifrado y descifrado así como generación de MACs basados en clave simétrica y funciones de firma digital para Mobile SET.
- La posibilidad de integrar WAP y SIM Toolkit, de modo que se pueda acceder desde WAP a aplicaciones desarrolladas según este último estándar, lo que permitiría acceder al SIM y utilizar funciones criptográficas y de seguridad.
- La creación de un estándar WPKI (Wireless PKI) que permita integrar la infraestructura criptográfica que define WTLS en infraestructuras de clave pública como las ya existentes en el dominio TCP/IP [MAÑA 99]. Los fabricantes de PKIs más importantes ya han puesto en el mercado diferentes productos de WPKI. Sin embargo, hasta que no se defina el estándar y los fabricantes de infraestructuras integren estos productos, su utilidad es bastante limitada. Entre los más destacados están los siguientes: Baltimore Technologies con Telepathy, la cual proporciona certificados de servidor, como ya se ha citado previamente. Certicom, propietaria de los algoritmos criptográficos basados en curvas elípticas ha presentado recientemente MobileTrust. Entrust ofrece Secure Wireless e-Business Solutions.

2.4 Propuesta.

Con el fin de solucionar este problema, existen las siguientes tres alternativas:

- Colocar la pasarela WAP en el extremo de la conexión del servidor Web, es decir, dentro de su misma zona de seguridad.
- Introducir una capa de seguridad sobre WAP, es decir, considerar a WAP meramente como un medio potencialmente inseguro de comunicación.
- Rediseñar el protocolo WAP para no utilizar la pasarela, empleando los estándares de Internet existentes.

WAP se diseña para convertir dos juegos de protocolos distintos, uno para redes con cables y otra para redes inalámbricas. De modo que la primera solución implica el uso inadecuado de protocolos en la red cableada. Y la tercera solución evita la optimización de protocolos para el ambiente inalámbrico. La solución que proponemos sigue la segunda estrategia. La consecuencia es que se pierden algunos de los beneficios provistos por la pasarela WAP [JUUL 01].

Esta propuesta consiste en una nueva capa de seguridad denominada por los autores WAE-Sec, compatible con TLS. La Figura 2.6. muestra la ubicación de esta capa dentro de la pila de WAP.

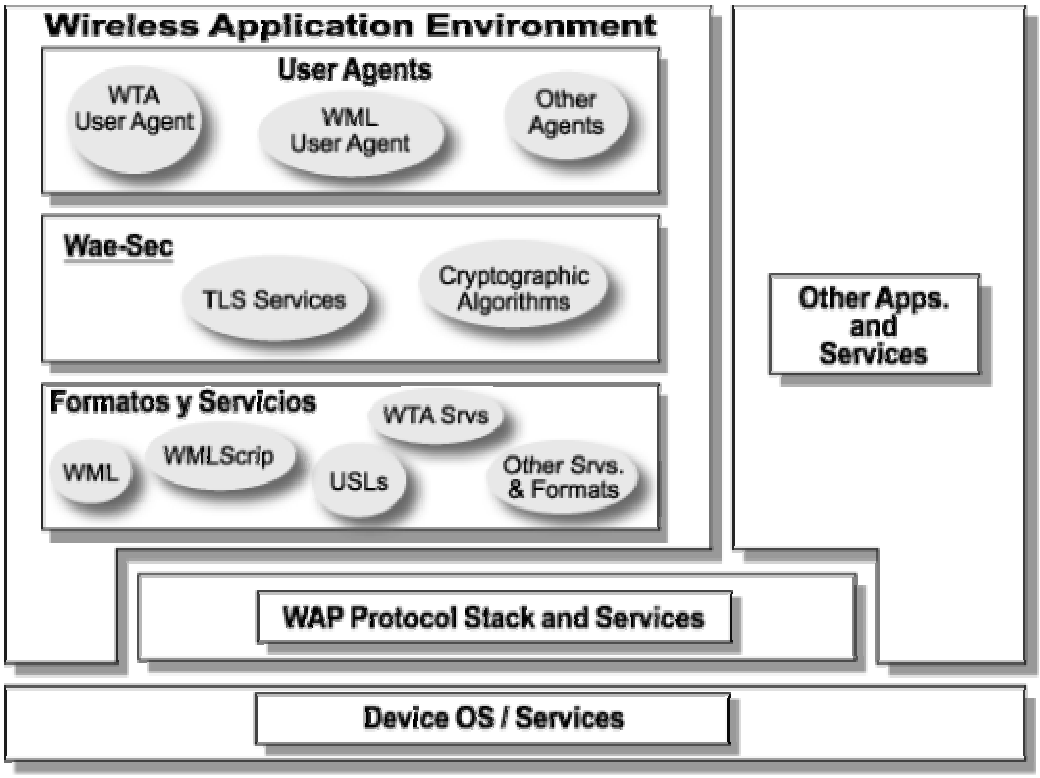


Fig. 2.6 Implantación propuesta de WAE-Sec..

WAP es una arquitectura de protocolos que permite el acceso a Internet mediante terminales móviles. El uso de las capas TLS[14] y WTLS[15] proporciona autenticidad, integridad y privacidad en los canales fijo e inalámbrico, pero el nivel de seguridad alcanzado no es suficiente para aplicaciones de comercio electrónico, ya que no se ofrecen servicios de seguridad extremo a extremo.

La propuesta de seguridad presenta la implantación de una nueva capa de seguridad dentro de la capa WAE[16], denominada WAE-Sec que permite la seguridad extremo a extremo, manteniendo compatibilidad entre TLS y WTLS.

Para los mensajes salientes es deseable que la pasarela WAP no intervenga en el descifrado con el fin de proveer un túnel seguro extremo a extremo. Si el cifrado se realiza tras la codificación WBXML, el servidor en la red Internet no será capaz de entender y traducir los tokens WBXML. En consecuencia, se debe evitar el uso de WBXML o proveer al servidor en Internet de los filtros que traduzcan los tokens a marcas WML o HTML. Hay un compromiso entre los requisitos de seguridad y la filosofía de un protocolo ligero para ambiente inalámbrico. La solución adoptada exige un incremento de tráfico, por las necesidades de seguridad.

Para superar los problemas existentes con WTLS y ofrecer a la comunidad de Internet la posibilidad de hacer negocios, de forma transparente y segura también en entornos WAP, se rediseña y reubica la capa de seguridad. La implementación de la capa de seguridad, se denomina por los autores WAE-Sec se implanta dentro de la capa de aplicación WAE de WAP, y se ubica en el lado del cliente WAP. WAE-Sec provee la seguridad extremo a extremo entre el cliente móvil y el servidor seguro de Internet, y simultáneamente evita la traducción en la pasarela de WAP.

Entre los beneficios tanto para el comprador como para los comerciantes on-line citamos los siguientes: a) El comprador dispondría de mayor oferta de sitios con los cuales hacer negocios.

---

[14] TLS = Transport Layer Security.

[15] WTLS = Wireless Transport Layer Security.

[16] WAE = Wireless Application Environment.

b) Los vendedores en Internet ya no necesitarían implantar infraestructura adicional ni convenios costosos con el operador de telefonía para establecer pasarelas especiales. c) Se eliminan las limitaciones que en un mercado global supone tener que realizar convenios con cada uno de los diferentes operadores y es quizás esta la causa de que hoy en día sean pocas las empresas que realicen comercio electrónico a través de WAP.

El problema radica en la ubicación de WTLS dentro de la pila del protocolo WAP, causando este hecho, entre otros problemas la traducción de la tokenización, traducción de códigos de alerta de TLS no compatibles, la doble compresión y descompresión en la pasarela WAP. Por tanto la nueva propuesta está obligada a cambiar de ubicación esta capa pero se plantea la incógnita de dónde puede ir esta capa de tal manera que sea lo suficientemente flexible y que a su vez aproveche el desarrollo hecho en WAP, además de permitir negociar extremo a extremo hacia cualquier servidor de contenidos de Internet.

Otra dificultad que se plantea en los algoritmos criptográficos, es el hecho que son una carga importante para la CPU y su tiempo de procesado impone restricciones; por tanto hay que estudiar de donde pueden los algoritmos criptográficos obtener apoyo de procesador, por ejemplo una CPU extra sería una posible solución.

En este momento se plantea la solución al tema de la seguridad: la capa WAE-Sec en detrimento de la capa WTLS. A continuación se plantearán las prestaciones que ofrece, luego su ubicación y finalmente las características generales que se han definido para la misma.

2.4.1 WAE-Sec.

La capa WAE-Sec ha sido desarrollada utilizando contribuciones de código de diferentes desarrolladores. Estas contribuciones incluyen el código fuente en lenguaje C denominado KA9Q[17] de Phillip Karn. La contribución de Brian Lantz denominada TNOS[18] que básicamente provee soporte de HTTP al KA9Q para entornos Linux y DOS [COME 95], el código WAP de Kannel [KANN 99], y el código SSL-TLS de Openssl [OPEN 98].

Para cumplir con los objetivos propuestos, se necesita modificar el control estándar de las funciones de codificación y control en el entorno WAP. Se incluyen las bibliotecas adicionales y algoritmos criptográficos a los servicios y formatos de WAE y se provee de un túnel TLS entre ambos extremos, se compatibilizan las alertas con TLS y realizan las pruebas de comunicación.

La modificación de las funciones estándar de WAP son las mínimas para proveer control sobre el túnel y evitar la intervención de la pasarela. Se busca compatibilizar TLS con WTLS. El sistema propuesto ofrece las siguientes prestaciones:

- *Seguridad extremo a extremo.*
- *Compatibilidad entre TLS y WTLS.*
- *Transparencia ante el usuario.*
- *Evita la Traducción y Descompresión en el nodo de la pasarela WAP.*
- *Evita la Doble compresión.*

La seguridad extremo a extremo entre el servidor de Internet y la terminal móvil es también indispensable para aplicaciones de comercio electrónico. La especificación de WTLS no proporciona este nivel de seguridad. El uso de WTLS y la seguridad a nivel de la capa de transporte TLS (Transport Layer Security) permite la privacidad en los canales inalámbrico e Internet, pero la seguridad alcanzada no es suficiente en aplicaciones de comercio electrónico; por ello se precisan mecanismos de seguridad extremo a extremo. Los autores proponen la implementación de una capa de seguridad nueva dentro de la capa Wireless Application Environment (WAE), denominada por los autores WAE-Sec. Esto hace posible la seguridad extremo a extremo, la compatibilidad con TLS, la transparencia ante el usuario, y evita la traducción y descompresión en el nodo de la pasarela de WAP. La sección 2 presenta brevemente el ambiente WAP y el modelo de seguridad de WAP. La sección 3 propone una alternativa a la arquitectura de la seguridad de WAP para resolver el problema mencionado.

---

[17] KA9Q = TCP/IP for amateur packet radio.

[18] TNOS = Tampa Network Operating System.

Prestaciones de WAE-Sec

Se propone realizar una nueva capa de seguridad que va a sustituir a la actual WTLS, intentando aprovechar, eso sí, las características beneficiosas que sí ofrece WTLS y sobretodo haciéndola compatible con el protocolo actual de seguridad en TCP/IP, el TLS, que ha sido aceptado como la capa seguridad a nivel de transporte por la comunidad de Internet.

Por tanto, el sistema propuesto ofrece las siguientes prestaciones:

- *Seguridad extremo a extremo:* La tokenización y compresión binaria se realiza a nivel de la capa de aplicación y consiste en la traducción de las marcas de WML por códigos que aligeran la carga a ser transportada por el canal inalámbrico en formato WBXML hacia y desde el móvil. La traducción del WBXML es realizada en forma relativamente simple mediante el uso de filtros que desmontan los códigos y los reemplazan por marcas WML o HTML del lado cableado de Internet. El problema surge cuando el tráfico que debe circular por la pasarela WAP está cifrado, pues la traducción de estas marcas implicaría poner en claro el contenido del mensaje, con el potencial riesgo de seguridad. Para evitar la traducción en la pasarela WAP existen varias alternativas:

    - a) Evitar la tokenización y compresión binaria, lo que implicaría un aumento de la carga en el canal inalámbrico. Las informaciones llegarían en formato WML/Script al móvil.

    - b) Proveer los filtros necesarios o interpretes de WBXML del lado del servidor en Internet, lo que es más adecuado pues también se aligera la carga en el tramo cableado de Internet y la traducción se realiza dentro de la misma zona de seguridad del extremo del servidor.

    - c) Negociar, de acuerdo a las capacidades del servidor, el formato de salida desde el terminal móvil, es decir, WBXML, WML/Script, o HTML. Lo que implica una carga adicional del canal inalámbrico y una carga de procesamiento en el extremo del móvil, que es el de menos recursos (de CPU, memoria, etc.).

- *Compatibilidad con TLS:* El protocolo TLS [DIER 99] actualmente es una propuesta de estándar de Internet ampliamente aceptado. El diseño de WTLS sigue prácticamente la misma filosofía, aunque con algunas diferencias debidas a las distintas características de los entornos. WLTS posee las siguientes dificultades:

    - *a)* La ubicación de la capa opcional WTLS dentro de la pila del protocolo WAP (entre WDP y WTP), impide el control sobre la compresión y codificación en las capas superiores, que no son deseables para proveer una transmisión segura extremo a extremo porque hacen necesaria la intervención de la pasarela WAP en el descifrado, la traducción y descompresión de la carga.

    - b) El entorno inalámbrico hace necesaria la introducción de códigos de alerta distintos a los de TLS. Para poder conseguir la compatibilidad extremo a

extremo, se deberían utilizar los códigos de alerta estándar de TLS e inhibir los códigos de alerta extendidos de WTLS que de todas formas eran traducidos en la pasarela WAP.

- c) Permitir el uso de otros algoritmos de adicionales cifrado como por ejemplo: RSA en lugar de las ECC exclusivamente. Afortunadamente en las especificaciones desde WAP 1.2 en adelante, ya se contempló este punto.

- d) El protocolo WTLS es prácticamente idéntico en su especificación al protocolo TLS, en las versiones 1.1 y 1.2 de WAP se la provee como una capa opcional, y por lo tanto, fácilmente modificable y reubicable dentro de la pila de WAP.

- e) Por estas razones, WAE-Sec se diseña compatible con TLS, y se ubica en la capa de aplicación de la torre WAP.

- *Transparencia ante el usuario*: Siguiendo la misma filosofía de TLS en Internet, en aras de la simplicidad de uso, se necesita ofrecer la seguridad de forma transparente al usuario. El agente del usuario (mini-navegador) debería manejar la seguridad de forma fácil y amigable al usuario.

  - *a)* Los agentes de usuario de la capa de aplicación WAE, implantan autenticación básica como se especifica en el RFC2068 HTTP 1.1 [FIELD 97], [FIELD 99].

  - *b)* El protocolo de sesión WSP utiliza el mismo juego de caracteres, cabeceras, y códigos de país del HTTP 1.1, de modo que es posible establecer un túnel extremo a extremo para transacciones seguras.

- *Eliminar la traducción y descompresión en el nodo de la pasarela WAP*. Evitar el uso de la descodificación y descompresión significa reducir la intervención de la pasarela WAP durante la transmisión de datos necesaria para garantizar seguridad extremo a extremo.

  - *a)* Esto sólo es posible reubicando la capa de seguridad en un nivel más alto, gracias a la compatibilidad de WSP con HTTP 1.1.

  - *b)* La capa de aplicación WAE ofrece facilidades en la sección de servicios y formatos, de modo que se permite la implantación de nuevos algoritmos criptográficos, en forma flexible.

  - *c)* Posterior al desarrollo e implantación de WAE-Sec (Junio 2000), la versión 2.0 de WAP (Junio 2001), provee de los algoritmos de cifrado en la capa de aplicación WAE. Otro desarrollo importante en Java, concretamente

JDK versión 1.4 integra los algoritmos de cifrado en esta versión (Junio 2001).

- *Doble compresión:* Evitar la tokenización WBXML, compresión en WTP y posterior cifrado y compresión con WTLS en este orden [WAPF 99], para luego descomprimir descifrar y traducir los mensajes en la pasarela WAP, además de ahorrar el coste computacional se reduce el riesgo del criptoanálisis. Si se obtiene el cifrado y compresión en la capa WAE-Sec y se aprovecha la compresión de WTP hasta la pasarela WAP sin exponer la información cifrada se produce un ahorro de recursos. Se debe mencionar que la portadora en GSM también ofrece servicio de compresión.

Estas son a grandes rasgos las prestaciones que va a ofrecer la nueva capa. Claro está que todas ellas persiguen un mismo fin que no es otro que el de proveer seguridad de una transmisión entre un terminal móvil compatible con WAP y un servidor seguro dentro de Internet. Partiendo de la premisa de que el estándar actual de seguridad TLS es totalmente seguro (siempre y cuando se consideren una serie de condiciones ya comentadas anteriormente), se puede afirmar, que, según el modelo propuesto se conseguirá un túnel seguro entre cliente y servidor.

Ubicación de la capa

La posición de la capa WAE-Sec, ya como su nombre propuesto indica reside en la capa superior de aplicación WAE. La principal razón es que al querer hacerla compatible con TLS, debe estar ubicada en una posición equivalente.

También es cierto que, si se quieren cumplir los requisitos anteriormente citados, y especialmente el principio extremo a extremo ésta es su única ubicación. Es decir, obviamente debe estar por debajo de la capa de aplicación, que es la que genera y recibe los datos útiles. Luego, si se quiere establecer un túnel seguro entre cliente y servidor, sin querer que una pasarela convencional WAP toque los datos ya hay que ponerlo por encima de WDP y finalmente si se quiere evitar la tokenización pues ya debe estar justo por encima de WSP. El hecho de tener que interactuar con la capa WAE hace que esté dentro de esta última.
En la Figura 2.6. se puede ver la ubicación final de la capa WAE-Sec dentro de la torre de protocolos WAP.

Los componentes de dicha capa son dos:

- Servicios TLS.
- Algoritmos criptográficos.

Características

La capa WAE-Sec es una combinación entre TLS y WTLS y por tanto tiene propiedades tanto de TLS como de WTLS. Del primero coge los algoritmos, la posición dentro de la torre de protocolos y el mecanismo de autentificación. Del segundo coge las propiedades que forman parte del entorno inalámbrico como pueden ser los códigos de alerta. De los dos adopta la estructura y los subprotocolos de los que está formado, que son idénticas en ambos casos.

Las características esenciales de la capa propuesta son prácticamente las mismas que las de la capa TLS proporcionando:

- Integridad de los datos: Se asegura que los datos intercambiados entre el terminal y un servidor de aplicaciones no han sido modificados y no es información corrupta.

- Privacidad de los datos: Se asegura que la información intercambiada entre el terminal y un servidor de aplicaciones no puede ser entendida por terceras partes que puedan interceptar el flujo de datos.

- Autentificación: Este protocolo contiene servicios para establecer la autenticidad del terminal y del servidor de aplicaciones.

- No repudio: Ninguno de las partes implicadas en una comunicación podrá negar su participación.

- Además se definen tres tipos de autentificación:
    - Autentificación del cliente y servidor.
    - Autentificación sólo del servidor.
    - Ningún tipo de autentificación.

Dicho protocolo está compuesto por dos capas: la primera correspondiente al protocolo de registro y sobre ésta, la segunda capa compuesta por los protocolos de Handshake, Alerta y Cambio de Cifrado.

El principal rasgo que ofrece con relación al modelo de seguridad actual es el de seguridad extremo a extremo. La pasarela adoptará aquí un papel pasivo, siendo su única tarea la de encaminar los paquetes de cliente a servidor y viceversa. La presencia de esta pasarela es necesaria ya que ha de pasar la información de un entorno inalámbrico a uno alámbrico y por tanto deberá modificar las capas bajas del protocolo para pasarlas de WAP a TCP/IP. Esta parte también funciona así actualmente. Sin embargo, el modelo actual también deja en manos de la pasarela la traducción de las capas más altas entre los dos protocolos y por tanto afectando a la capa de seguridad. La pasarela ve venir los paquetes los descomprime los traduce y los vuelve a comprimir. Esto se considera como una debilidad importante ya que esta compresión/descompresión deja los datos al descubierto durante unos instantes dentro de la pasarela, que en principio es propiedad de las operadoras telefónicas y no de empresas de seguridad. El sistema propuesto, por tanto, evita esta compresión/descompresión creando prácticamente un túnel entre cliente y servidor, quitando un punto débil:

En lo que se refiere a la torre de protocolos, la capa WAE-Sec se comporta tal y como lo hacen las demás capas de la arquitectura WAP.

Varios artículos y publicaciones recientes están sosteniendo la idea adoptada en esta tesis; proporcionando una seguridad extremo a extremo haciendo una capa de seguridad como la aquí diseñada y explicada [THAN 00],[WAPF 01].

Quizás puede ser muy osado aplicar algoritmos como RSA dentro de este tipo de entornos. Sin embargo, cabe la posibilidad de incrementar la biblioteca criptográfica con algoritmos adicionales a la ECC ya que en el estándar WAP 1.2 ya se contempla incorporar RSA pese a su costo computacional.

2.5 Conclusiones.

Se ha presentado el protocolo WAP y cómo WTLS implementa los servicios de seguridad necesarios.

El desarrollo de servicios de comercio electrónico en la actualidad utilizando WAP tiene una serie de problemas de seguridad. Estos problemas son:

- Autenticación extremo a extremo.
- Confidencialidad en la pasarela WAP.

En consecuencia, proveer un túnel seguro entre WAP y TCP/IP mediante el uso de TLS es factible y necesario. WAP proporciona una codificación ligera y las funciones de compresión deseables para ambientes inalámbricos pero se contraponen con los requisitos de seguridad criptográfica.

A lo largo del capítulo se ha presentado una solución que ofrece compatibilidad extremo a extremo con TLS y localiza la capa dentro de WAE desde donde es capaz de controlar la forma en que los datos son transmitidos para evitar la intervención de la pasarela WAP.

WAE es una capa adecuada por la flexibilidad que brinda para la adición de nuevas bibliotecas y algoritmos criptográficos como parte de los servicios y formatos de esta capa.

Esta solución permite conseguir además de seguridad extremo a extremo;

- Compatibilidad con TLS.
- Transparencia ante el usuario.
- Evitar la traducción y descompresión en la pasarela.
- Evita la doble compresión.

Capítulo 3

Implantación de la seguridad extremo a extremo en WAP con WAE-Sec.

3.1 Introducción.

En el capítulo anterior se mostró la debilidad que ofrece WAP en cuanto se refiere a seguridad. La arquitectura WAP se diseñó a partir de TCP/IP y de los estándares de Internet para soportar adecuadamente las comunicaciones en entornos inalámbricos. WAP puede prescindir de la capa IP y por tanto IP-Sec no queda contemplado. La solución para la seguridad que propone WAP es WTLS; este protocolo presenta algunas diferencias que lo vuelven incompatible con TLS, y las ya expuestas en el capítulo anterior:

- No se provee autentificación extremo a extremo.
- Es vulnerable dentro de la pasarela WAP, ya que para la traducción de WTLS a TLS quedan en claro los contenidos.

Varios autores consideran que esto se podría solucionar trasladando las capas altas de la pasarela WAP hacia la zona de seguridad del servidor de comercio electrónico. Actualmente la pasarela WAP pertenece a las operadoras telefónicas. Otra propuesta considera al WAP meramente como un protocolo inseguro, y plantea la seguridad sobre la capa de aplicación, con lo que se desaprovecha las cualidades del protocolo WAP [JUUL 01]. No todos los negocios en Internet están en posibilidad de contratar un acceso especial con cada proveedor telefónico. Además, muchos ya han hecho cuantiosos esfuerzos para hacer negocios en Internet mediante TLS y desean aprovechar esta infraestructura y adaptarla al nuevo entorno WAP sin necesidad de inversiones adicionales, por lo que se debe adoptar otra solución.

A lo largo de este capítulo se describirá la solución implementada, sus prestaciones, limitaciones y los resultados obtenidos.

3.2 Implicaciones de WAE-Sec.

Se ha intentado que la implantación de la capa de seguridad WAE-Sec tenga el menor impacto posible en la arquitectura de WAP. Sin embargo, la implantación de esta capa exige los siguientes cambios en cada uno de los participantes:

a) *Cliente:* Se inserta la capa de seguridad compatible con TLS en WAE, con las prestaciones citadas en el capítulo 2.
b) *Pasarela:* Se evita utilizar la capa WTLS del WAP. Se evita la intervención de la pasarela en la descompresión y traducción de los contenidos a nivel de la capa de aplicación.
c) *Servidor:* Se procura mantener completa compatibilidad con los servidores de Internet que soportan TLS. Por lo tanto, se procura que no existan cambios en este participante.

3.3 Desarrollo de la capa WAE-Sec.

Como se ha comentado en el punto anterior, solo debe modificarse el cliente WAP. El cliente se comunica a través de una pasarela WAP con un servidor seguro en Internet. Se ha utilizado el código de un cliente https genérico utilizando las librerías del OpenSSL de Apache [OPEN 98]. OpenSSL es una contribución de software a código abierto y de libre distribución para ambiente operativo Linux, que proporciona los mecanismos de negociación de Secure Sockets Layer versión 3.0 y de TLS versión 1.0 (SSL 3.1), con sus algoritmos criptográficos, protocolos de seguridad, (protocolos de registro, handshake, alerta y reanudación).

La construcción de la torre WAP sobre sistema operativo Linux y MS-DOS se realizó a partir del paquete KA9Q [KARN 93a] una torre TCP/IP a código abierto en lenguaje C adaptada al entorno de radio y PC.

3.3.1 Desarrollo del Cliente.

Con el fin de implantar la capa de seguridad compatible con TLS dentro de la capa de aplicación WAE, se utiliza OpenSSL, se modifica la entrada y salida con el sistema operativo redirigiendo su comunicación hacia la implantación a código abierto de la torre WAP y se efectúan modificaciones al código de la capa de registro de TLS. La implantación de la torre WAP se realiza en un entorno de PC con Linux. Las funciones propias del OpenSSL dialogan con el sistema operativo Linux a través de la torre WAP para empaquetar las tramas que por defecto están activas. El cliente WAP acepta el paso de argumentos como son:

- La versión del protocolo a utilizar (TLS v.1.0).
- El certificado a utilizar en caso de que el servidor lo pida.
- El nombre del servidor así como el puerto al que se conecta (por defecto HTTPS en el puerto 443).
- Parámetros de control y la salida de estadísticas del programa.

Una vez recogidos los parámetros e inicializadas las variables necesarias se crea la comunicación mediante protocolo WAP y realiza el handshake para establecer los parámetros criptográficos de la sesión, se identifica el servidor así como su certificado y se identifica el mismo si es necesario. Tras el establecimiento de la comunicación se realiza una conversación con el servidor, se envía el comando “GET /pagina.html HTTP1.0\n\n” y se recibe la página.html.

Como salida nos muestra en un fichero todas las características criptográficas de la sesión así como todo el texto cifrado recibido, y en otro la página.html descifrado. Esta última es enviada a un navegador Web para su visualización.

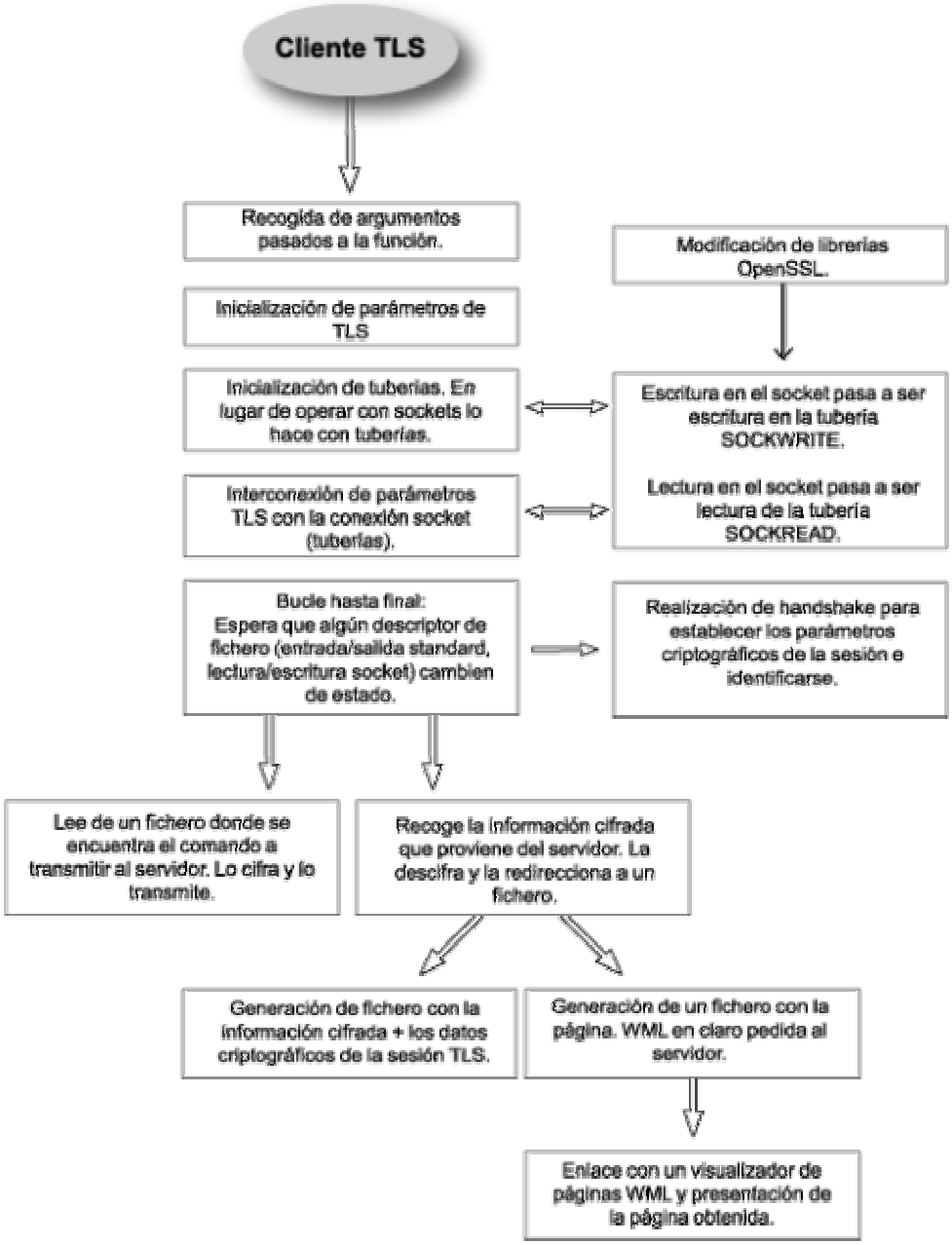


Fig. 3.1 Cliente TLS.

3.3.1.1 Diseño del componente TLS de la capa WAE en el cliente WAP.

El diseño de este componente se basa en OpenSSL que es una librería de programación con código fuente de libre distribución que implementa y proporciona servicios de TLS. Esta librería proporciona tanto servicios de criptografía (cifrado de clave simétrica, clave pública, hash, certificación X.509, ...) como servicios específicos sobre el protocolo TLS; como podría ser la manipulación automática de sockets. Esta librería se basa en SSLeay, desarrollada originalmente por Eric A. Young y Tim J. Hudson, con la contribución de otros programadores sobre la librería ya establecida creando OpenSSL.

Se ha creado un programa cliente TLS partiendo de las librerías anteriormente compiladas. Este programa es el encargado de gestionar el protocolo TLS, es decir, de inicializar variables, hacer el handshake completo y cifrar/descifrar los mensajes entre cliente y servidor. Se han programado las rutinas para extraer hacia un fichero todos los datos de la negociación (*handshake*) en binario así como los datos de la sesión y finalmente los datos pedidos al servidor (en nuestro caso la página WML) tanto en claro como cifrados, que luego son visualizados en el navegador WAP construido para tal efecto.

El programa cliente se encarga también de comprobar los datos del certificado del servidor. Se comprueba que tiene el formato estándar y presenta por pantalla un mensaje con los datos del certificado donde se da opción al usuario de fiarse o no del servidor antes de continuar el proceso. En la Fig. 3.1 se puede observar el esquema de bloques funcionales del programa.

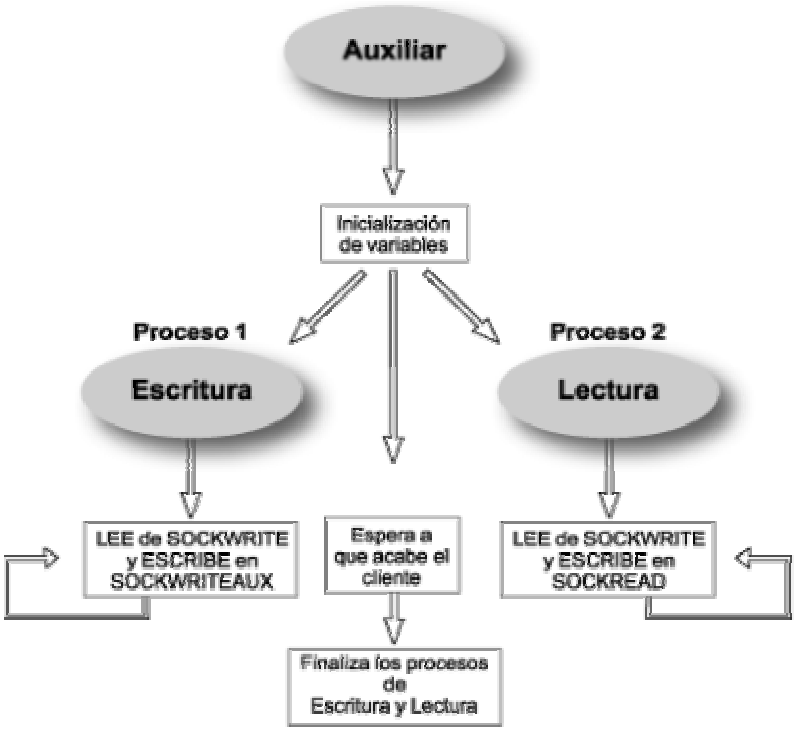


Fig. 3.2 Programa auxiliar.

Programa auxiliar

El programa auxiliar permite redirigir las salidas/entradas del cliente TLS. Su función es hacer de programa intermedio entre los dos: lo que le envía uno lo recoge y se lo pasa al otro en ambas direcciones como se puede ver en la Fig. 3.2.

WAE

Es el programa principal encargado de llevar a cabo las tareas de la nueva capa WAE resultante. Una vez añadida la subcapa WAE-Sec se tiene que encargar de juntar todas las subcapas y crear una única capa. Ésta se comunicará con las capas inferiores de la torre WAP, tanto para transmitir como para recibir de ellas la información correspondiente.

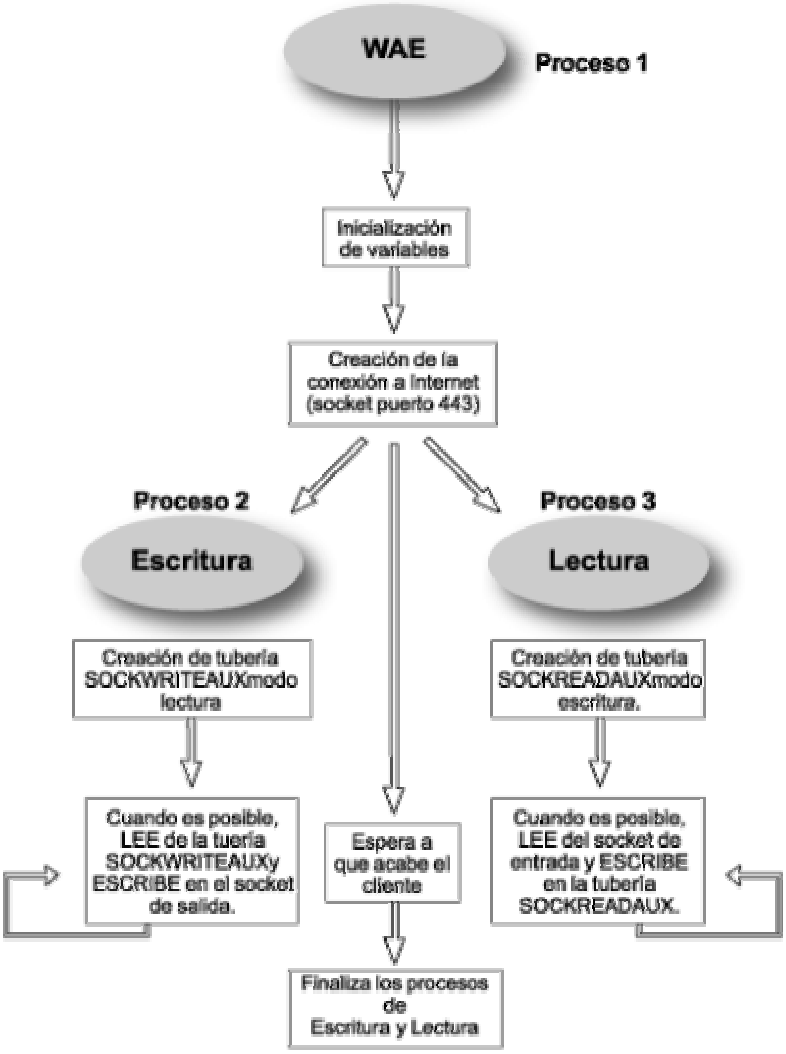


Fig. 3.3 Programa WAE.

Ya que el puerto 443 es el puerto por defecto para las conexiones seguras por Internet con TLS y el programa va a intentar conectarse con un servidor que tendrá activo ese puerto, se debe abrir un socket con dicho identificador de puerto. A nivel de la capa de aplicación, recibe la información del programa auxiliar y la transmite a la que sería la pasarela WAP equivalente. Esta información, al estar cifrada a este nivel, no sufrirá ningún cambio en la pasarela y llegará a destino tal y como se envía desde aquí. En sentido inverso sucede lo mismo. El esquema de bloques de este programa se puede ver en la figura 3.3.

Integración de la capa WAE-Sec en el cliente WAP.

Para la realización del cliente se ha utilizado una de las variantes del KA9Q denominada TNOS [LANT 96], que es la versión modificada del anterior con algunas mejoras. Está disponible tanto para el sistema operativo D.O.S. como para Linux y ya que el cliente WAE está construido bajo Linux, las capas inferiores también habrá que ejecutarlas en este mismo sistema operativo.

TNOS es una aplicación multi-hilo (*multi-thread*) que contiene una implementación completa del protocolo TCP/IP con manejadores (*drivers*) para las capas inferiores. Soporta entornos de radiofrecuencia, fundamentalmente en entorno Amateur Packet Radio. TNOS funciona como pasarela, un router, firewall, cliente/servidor de: e-Mail, FTP, HTTP, Telnet,

etc., es mantenido por su propio autor, Brian A. Lantz - KO4KS, y diversas listas de distribución.

Implementación y configuración.

Una de las principales dificultades ha sido modificar TNOS, la torre de protocolos para pasarla de TCP/IP a la especificación de WAP. Siguiendo el modelo OSI, e interviniendo las comunicaciones entre capas con las SAP, (Service Access Poínos) con colas de espera finitas entre ellas, se ha conseguido la independencia de las mismas y el manejo de los datos de la aplicación como paso de capa por capa hasta llegar a la física y añadir al paquete la cabecera correspondiente. Entre las principales modificaciones están las siguientes:

- Modificación del protocolo TCP para convertirlo en WTP/WDP.
- Creación de capa WSP por encima de TCP (WTP) correspondiente a la torre de protocolos WAP.
- Creación de capa superior a WSP (llamada WAP) equivalente a WAE.
- Adecuación de la capa WAP (WAE) para comunicarse adecuadamente con el cliente WAE implementado.
- Paralelización de procesos dentro de la capa WAP (WAE).

Es necesario comentar que según los estándares WAP existe la posibilidad de poner IP por encima de la portadora. Por tanto, y adoptando como nuestra portadora el PPP (también según especificaciones se puede utilizar cualquier portadora) se ha utilizado por encima IP. WDP utilizado sobre IP se convierte en UDP. Las principales funcionalidades de TCP con las correspondientes modificaciones equivalen a la capa WTP, adecuada con las especificaciones WAP. En el entorno de prueba, las capas inferiores a la WAE son una herramienta para poder probar la capa de aplicación, y la implementación de las capas propias del WAP en un entorno inalámbrico (WSP/WTP/WDP/GSM) queda en un segundo lugar en este trabajo.

Por encima de esta capa WTP/WDP se ha implementado la capa WSP, compatible con HTTP en Internet. Esta capa es la encargada de gestionar la sesión WAP así como de hacer las peticiones de la capa de aplicación y recibirlas. Por encima de WSP se ha implementado la interfaz con la capa de aplicación WAE. Para poder realizar la comunicación con la otra aplicación se han paralelizado los procesos de lectura y escritura dejando en dos funciones (hilos) cada una de estas dos tareas comunicando entre ellas los dos procesos que lanzan la aplicación WAE.

En cuanto a la traducción entre HTML y WML (que especifica el protocolo WAP que debe hacer la pasarela WAP), el hecho de haber establecido un túnel seguro entre cliente y servidor hace imposible este hecho. Además la información que viaja en ningún caso sería HTML sino WML cifrado y por tanto, la pasarela, que no tiene ningún tipo de información sobre la cifrado que se está utilizando no podría saber cuál es el código HTML que se está transmitiendo.

Las principales funcionalidades de las que se dotó a la pasarela fueron las siguientes:

- Router.
- Traductor de protocolos.
- Capturador de paquetes en las dos direcciones

Una vez explicadas las razones se estudiarán las funcionalidades de las que se le dotaron. En primer lugar se encarga de pasar los datagramas que le llegan del cliente al servidor (siempre y

cuando las cabeceras IP del cliente indican que quieren una petición de dicho servidor) y viceversa. Sus tablas de enrutamiento lo dicen así como se podrá ver en el apartado siguiente.

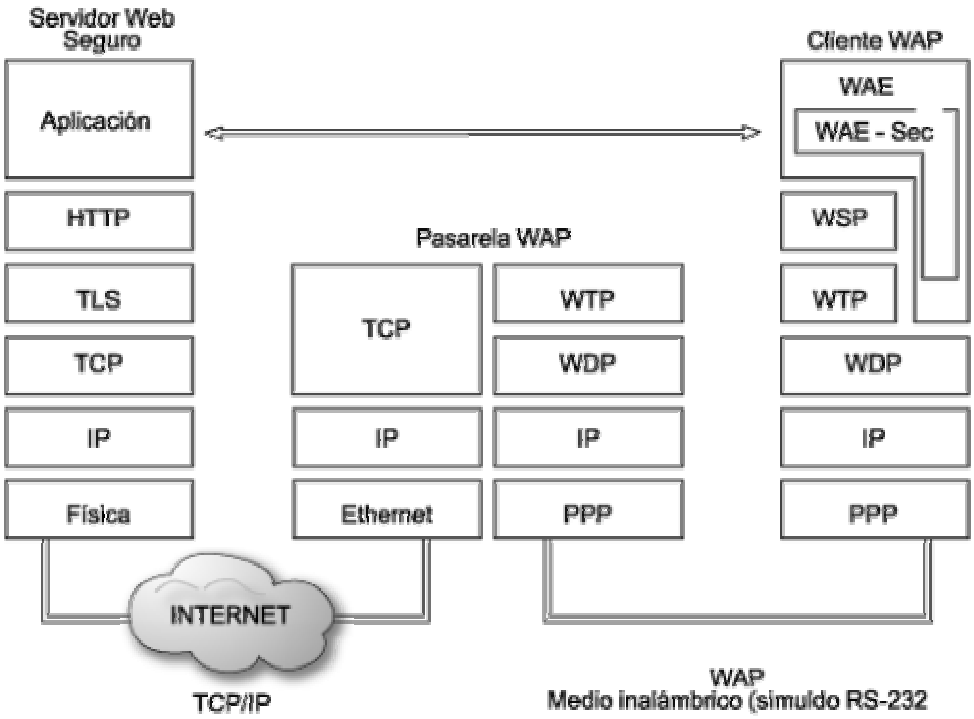


Fig. 3.4 Traducción de protocolos en la pasarela.

La última función de la que se ha dotado ha sido la de la captura de los paquetes que viajan en ambos sentidos. Esta no es la tarea de una pasarela convencional pero se creyó necesario para analizar la información que viaja por la red y para poder verificar el correcto funcionamiento del sistema. También se creó una función para almacenar en un archivo *.log* esta información.

En el siguiente apartado se podrán ver estos resultados para una comunicación segura completa: handshake, petición página WML y recepción de página WML.

3.4 Implantación de la capa WAE-Sec.

Como se ha comentado en el punto anterior, debe modificarse el cliente WAP. El cliente se comunica a través de una pasarela con un servidor seguro en Internet, como muestra la figura 3.5. Para realizar la capa de seguridad del cliente TLS en la capa de aplicación WAE de la torre WAP (la que se ha denominado WAE-Sec), se ha utilizado el código de un cliente https genérico utilizando las librerías del OpenSSL de Apache.

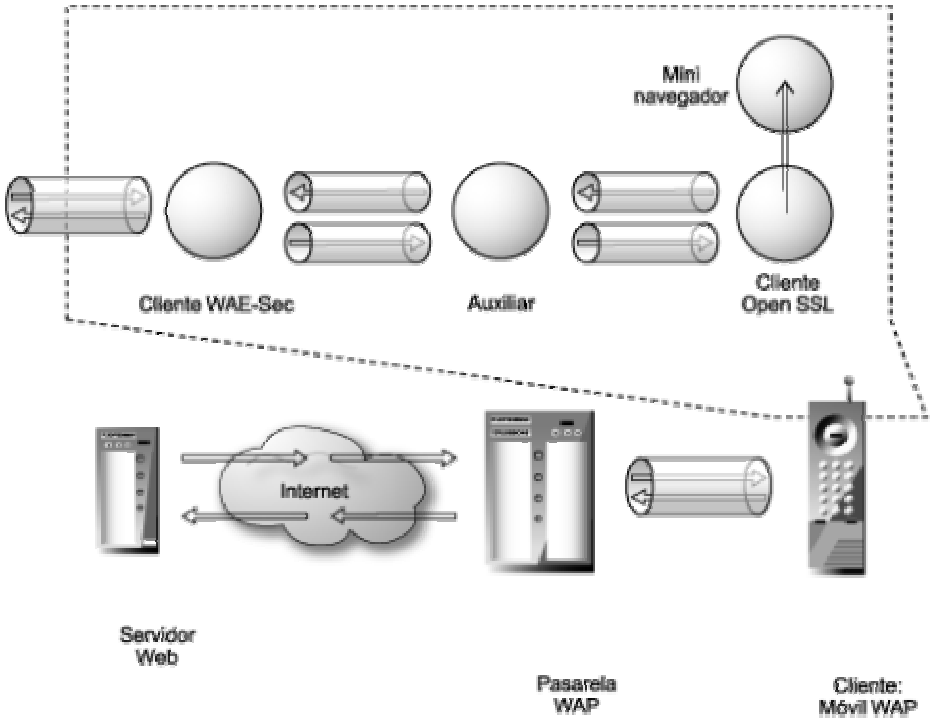


Fig. 3.5 Esquema del cliente.

Escenario de pruebas.

Para poder realizar las pruebas pertinentes era necesario emular un cliente WAP con una torre de protocolos adecuada como también generar un entorno de simulación lo más parecido al WAP real.

Asumiendo que el esquema WAP está formado por cliente-pasarela-servidor, se ha desarrollado un entorno igual con el mismo flujo de datos aunque cambiando algunas de las funcionalidades de la especificación de WAP.

En todo momento lo que se está buscando es crear un túnel que supere esa pasarela ya que es en ella donde surgen la mayoría de los problemas de seguridad existentes actualmente en la tecnología WAP. Los estándares WAP obligan a la presencia de dicha pasarela por lo que se ha tenido que diseñar un sistema simulado muy semejante al real. En caso de no existir la pasarela, el cliente, teléfono móvil se intercambiaría información directamente con el servidor y entonces sí que existiría ese túnel con seguridad extremo a extremo.

El sistema general se ha desarrollado de forma independiente al resto de la red corporativa aislando las tres máquinas implicadas ya que era necesario para realizar las pruebas correspondientes en el ambiente simulado y sacar los resultados y conclusiones pertinentes. Este aislamiento no condiciona de ninguna manera las pruebas realizadas ni la veracidad del sistema. Dado que no se percibe ningún requisito especial en el servidor se ha trabajado sobre un estándar basado en Apache y Mod_Ssl cuya configuración no será objeto de descripción.

Desarrollo del entorno de trabajo.

Para poder realizar las pruebas pertinentes era necesario tanto emular un cliente WAP con una torre de protocolos adecuada como generar un entorno simulado lo más parecido al real WAP. Asumiendo que el esquema WAP está formado por cliente-pasarela-servidor se ha desarrollado un entorno igual con el mismo flujo de datos aunque cambiando algunas de las

funcionalidades de las especificaciones WAP [WAPF 99] para poder adecuar el trabajo realizado al entorno.

El hecho de no disponer de un terminal móvil para realizar las pruebas ha condicionado la generación del sistema de pruebas teniendo que sustituir el radio enlace por una conexión por puerto serie como se verá en las próximas secciones

En la figura 3.6 se pueden ver las máquinas intervinientes, sus direcciones y los protocolos entre ellas además de la comparación con el sistema real.

Para poder realizar el sistema se ha diseñado una red privada de clase C (192.168.x.x). El encargado de encaminar los paquetes de cliente a servidor es el que hace de pasarela WAP ya que debido a que no se disponía de otra máquina que albergara un servidor de dominios y unas tablas de enrutamiento ha habido que configurarlas manualmente en cada máquina y especialmente en la del medio para que las tres se pudieran comunicar. Desde la pasarela hasta el servidor Web el camino es el mismo que el que se recorre según especificaciones WAP, es decir, TCP/IP puro. Sin embargo, entre cliente y pasarela se ha cambiado el radio enlace por un cable con RS232 mediante el protocolo PPP (Point to Point Protocol) a 9600 bps que es al menos la velocidad de la portadora actual GSM en entornos móviles.

3.4.1 Implantación del cliente WAP.

Para poder realizar una torre de protocolos compatible con WAP, independiente de las capas superiores y que se comunicase con la capa WAE implementada y explicada en el capítulo 2, hubo que realizar algunos cambios al paquete y recompilarlo posteriormente.

Primero de todo se adecuó la capa TCP para que fuera compatible con WAP. Es necesario comentar que según los estándares WAP existe la posibilidad de poner IP por encima de la portadora. Por tanto, y adoptando como nuestra portadora el PPP (Point to Point Protocol) (también según especificaciones se puede utilizar cualquier portadora) se utilizará por encima IP. Y por encima de IP WAP acepta UDP, es decir, que el WDP utilizado sobre IP se convierte en UDP. Además, ya que las principales funcionalidades de TCP equivalen a la capa WTP pero también tiene parte de UDP que a su vez equivale a WDP, se ha convertido la capa TCP existente en una especie de capa WTP/WDP adecuada para el entorno de pruebas y totalmente compatible con especificaciones WAP. No hay que dejar de lado que se trata de un ambiente simulado sobretodo teniendo en cuenta la capa inferior que en vez de ser un radio enlace es un cable que une cliente con pasarela. Además, las capas inferiores a la WAE son una mera herramienta para poder probarla, y la implementación de las capas propias del WAP en un entorno inalámbrico (WSP/WTP/WDP/GSM) queda totalmente fuera del abarque de este proyecto ya que, entre otros factores, no se disponía de un teléfono móvil.

Por encima de esta capa WTP/WDP se ha implementado la capa WSP, muy parecida a la HTTP en WWW. Esta capa es la encargada de gestionar la sesión WAP así como de hacer las peticiones de la capa de aplicación y recibirlas.

Y por encima de WSP se ha implementado una interfaz con la aplicación WAE desarrollada anteriormente en el capítulo 2, es decir, una serie de funciones que se comunican con la WAE y por tanto hacen sus funciones.

Para poder realizar la comunicación con la otra aplicación se han paralelizado los procesos de lectura y escritura dejando en dos funciones (hilos) cada una de estas dos tareas y comunicándose entre ellas con los dos procesos que lanzaba la aplicación WAE.

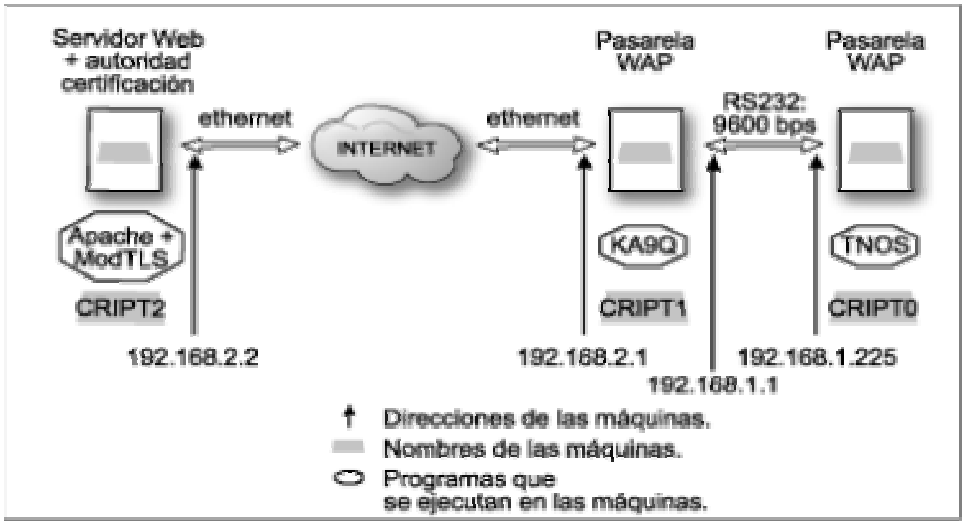


Fig. 3.6 Entorno de emulación.

3.4.2 Implantación de la pasarela.

Configuración.

El archivo de configuración es bastante extenso haciendo especial hincapié en el protocolo de enrutamiento así como de sus tablas. En principio el paquete KA9Q dispone de este servicio pero por extrañas razones no funcionó tal y como era de esperar. Después de ligeras modificaciones se consiguió el correcto funcionamiento.

Para configurar esta pasarela como Router fue necesaria seguir las especificaciones de un protocolo de enrutamiento existente. Actualmente existen dos opciones para llevar a cabo el enrutamiento: el estático y el dinámico. Una vez realizadas todas estas capas ha habido que comunicarlas entre ellas pasándose los mensajes correspondientes. El esquema de capas de protocolos queda según se muestra en la figura 3.7.

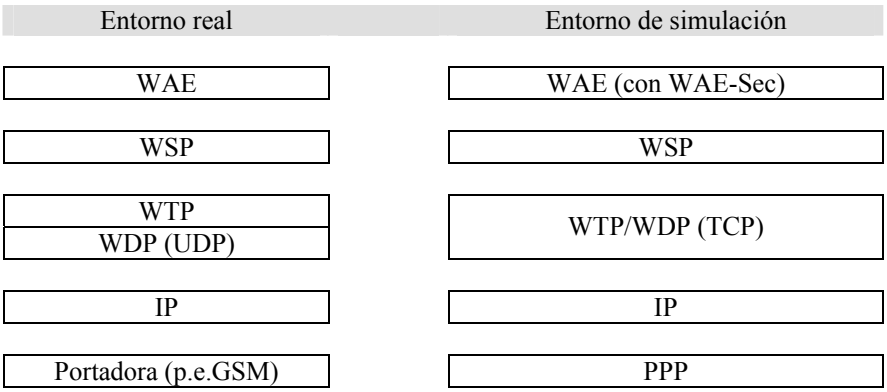


Fig. 3.7 Torre de protocolos emulada.

Otra de las tareas que se han realizado ha sido la de la configuración del entorno para que el cliente se comunicara adecuadamente con la pasarela. En un entorno real WAP y según información de las empresas que ofrecen este servicio actualmente, el proceso de configuración

del aparato móvil es complicado ya que las pasarelas todavía no se han desarrollado en su totalidad además de existir pocos centros que acepten dichos servicios.

Para poder simular un entorno WAP con la presencia de una pasarela se tuvo que realizar la conexión entre los dos mediante algún mecanismo que simulara un radio enlace a unos 9600 bps que es lo que ofrece la portadora actual de los dispositivos móviles GSM. Se decidió utilizar el protocolo PPP mediante un cable RS-232 conectado directamente entre los dos. En cuanto a la velocidad de transmisión los dos sistemas son iguales. Sin embargo, en cuanto a errores y ruido de la transmisión son muy diferentes porque un radio enlace es muy propenso a tenerlos mientras que un enlace punto a punto es todo lo contrario siendo los errores inexistentes. De todas formas, la manera de comunicar las máquinas no condiciona en absoluto la realización y comprobación del proyecto ya que como se ha dicho anteriormente es una herramienta y no un fin.

Por tanto, se configuró la aplicación TNOS para que funcionase correctamente con PPP. También se generó un Script para empezar la comunicación entre las dos máquinas (de hecho, se generó un par, uno para cliente y otro para pasarela). Estos archivos se ejecutan al final del programa de arranque de ambas aplicaciones, pasándose una serie de mensajes, quedando comunicados desde entonces. En la tabla 3.1 se puede ver la configuración resultante del PPP (llamado dentro de la aplicación pp0) con un ejemplo después de haber establecido una comunicación con al servidor.

Pp0	IP addr 192.168.1.225 MTU 1500 Metric 1 Link encap PPP Flags 0x0 trace 0x0 netmask 0xffff0000 broadcast 192.168.255.255 Sent: ip 1 tot 6 Recv: ip 17 tot 23						
Network Protocol Phase							
	2810 In	22 Flags, 0 ME, 0 FE, 0 CSE, 0 Other 2 LCP, 0 Pap, 4 IPcp, 0 unknowns					
	124 Out	5 Flags, 0 ME, 0 Fail, 2 LCP, 0 Pap, 4 IPcp					
LCP Output		MRU	ACCM	AP	PFC	AcFc	Magic
	Remote	1500	-0xffffffff	None	No	No	Unused
	Local	1500	+0x00000000	None	No	No	Unused
PAP Closed							
	Mesagge: ‘none’						
IPCP Opened							
	Local IP address 192.168.1.225 Remote IP address 192.168.1.1						
	In TCP header compression enabled, slots=4, flag = 0x01						
	Out TCP header compression enabled, slots=4, flag = 0x01						

Tabla 3.1 Configuración del PPP del cliente.

También se tuvieron que realizar manualmente las tablas de enrutamiento (tabla 3.2) por el hecho de no disponer de un servidor de dominios para realizar esta tarea. Se pueden observar los comandos necesarios para llegar a ellas.

Destination	Len	Interface	Gateway	Metric	P	Trace	Use
192.168.0.0	16	Pp0	192.168.1.225	1	/	Man	0
192.168.255.255	32	Pp0	192.168.1.225	1	P	Man	0
Default	0	Pp0	192.168.1.225	1	/	Man	0

Tabla 3.2 Enrutamiento del cliente.

3.5 Resultados.

3.5.1 El cliente WAP.

Configuración.

Tal como se ha dicho para poder simular un entorno WAP se tuvo que realizar la conexión entre el cliente y la pasarela mediante un mecanismo que simulara un radio enlace a 9600 bps que es aproximadamente lo que ofrece la portadora actual de los dispositivos móviles GSM.

Resultados en el Cliente.

Para comprobar el correcto funcionamiento de la capa WAE-Sec en el cliente WAP, se han realizado una serie de pruebas con peticiones de páginas WML y de archivos WBMP con y sin cifrado.

El primer caso es la petición de página WML segura. Con la capa WAE-Sec integrada en el cliente WAP observamos el código fuente de la página (tabla 3.3):

```
<?xml version="1.0"?>
<!DOCTYPE wml >


<wml>
  <card id="init" newcontext="true">
    <p align="center">
      <b>PRUEBAS WAP.</b><br/>
      Departamento de Matemática aplicada y Telemática.Por<i> Soriano-Ponce-Mur </i>
.Esta es una página de prueba de la capa <small>WAE-Sec</small>.
<br/>       Esta pagina ha sido pedida por el puerto 443 al servidor
cript2.upc.es
      <br/>
      <a href="#more">Autores WAE-Sec</a>
    </p>
  </card>

  <card id="more">
    <do type="prev" label="back">
      <go href="#init"/>
    </do>
    <p align="center">
      Soriano-Ponce-Mur </p>
  </card>
     </wml>
```

Tabla 3.3 Página WML en claro.

Es una pequeña página que consta de un texto con letra en itálica, negrita y pequeña además de un link. Luego de la correspondiente negociación, petición y respuesta con TLS, se ha obtenido la siguiente página cifrada:

```
Read from 080DBAC8 [080E15B5] (960 bytes => 960 (0x3C0))
0000 - 83 0c 94 16 26 d1 36 e8-d3 8d 93 21 32 32 42 82   ....&.6....!22B.
0010 - e4 87 ec c6 ca d0 dc 45-17 1f 01 49 ee 18 40 de   .......E...I..@.
0020 - de da 53 1e 35 8f e2 1e-61 fe 81 86 b9 84 0a 79   ..S.5...a......y
0030 - 35 6f 13 f1 35 74 b4 8c-75 9a ef 3f a6 8a 7f c7   5o..5t..u..?....
0040 - 2b 52 17 b5 2d 98 77 f6-46 c8 1a be 05 de d7 13   +R..-.w.F.......
0050 - 11 44 6e 92 0e 45 a7 43-ec 43 ae 3d 95 da 02 aa   .Dn..E.C.C.=....
0060 - 5f 2e 48 cf 80 89 9b 81-fa cd bf 46 76 35 0c 30   _.H........Fv5.0
0070 - 7e 0a c8 94 51 db af 79-77 a6 e2 c3 26 68 86 36   ~...Q..yw...&h.6
0080 - 3f 0d 74 cb c0 14 5c 8c-3d 43 2a 23 e8 f0 4f 74   ?.t...\.=C*#..Ot
0090 - 28 09 f7 cf 62 06 a7 d9-cc 42 38 2d f6 47 d2 d5   (...b....B8-.G..
00a0 - 31 95 85 14 7c b8 21 a9-af 43 95 df e8 0e f2 0f   1...|.!..C......
00b0 - 3f c7 5b 0c 94 06 d1 16-55 a8 46 7c da 49 ab 5f   ?.[.....U.F|.I._
00c0 - d1 55 13 a8 5b 66 f6 33-8d 14 1b ea db 83 17 7d   .U..[f.3.......}
00d0 - 7f 59 f6 84 d4 49 02 08-16 d0 e3 9e a4 60 77 97   .Y...I.......`w.
00e0 - 77 8a 10 f5 6b 13 b6 9c-62 38 c8 15 71 5b ba fd   w...k...b8..q[..
00f0 - 6f bd 42 c5 52 b7 ce df-e5 71 30 c2 bc ca e7 e3   o.B.R....q0.....
0100 - b4 ff 85 72 2c 07 1f ee-7f 5f 9b 79 16 73 34 08   ...r,...._.y.s4.
0110 - 3f 77 75 40 ad ec 75 21-96 a8 bb 75 ca c9 1f 3f   ?wu@..u!...u...?
0120 - dc e6 d5 6a a8 f4 cd 66-31 e5 52 73 5b d5 c8 b0   ...j...f1.Rs[...
0130 - 09 a9 cd 5c 09 3c 03 9e-9c 54 1c fa 6b 10 91 de   ...\.<...T..k...
0140 - 35 48 de 56 86 fe f6 34-cc 1c 02 64 7e 0f 34 d6   5H.V...4...d~.4.
0150 - 55 45 d7 94 b6 aa 88 74-9e fc fd ca 43 59 7a 03   UE.....t....CYz.
```

```
0160 - 24 c5 43 23 9b 5b 09 23-4e 15 cd cf 03 6f 51 01   $.C#.[.#N....oQ.
0170 - 6b 86 35 a0 e2 84 5a 42-91 af 17 62 99 84 b5 13   k.5...ZB...b....
0180 - 05 e8 3c 5d 39 7b 57 45-15 4e 14 12 f1 02 2f 5e   ..<]9{WE.N..../^
0190 - 48 7d 04 b6 aa 02 b1 1d-ae ca 62 ef b8 05 27 ee   H}........b...'.
01a0 - f6 ea 94 8c 5c 64 e6 c2-38 99 25 ea e4 a6 06 83   ....\d..8.%.....
01b0 - de 6a 52 55 cb ab dc 4d-2b 0e 8c c2 b4 5d 72 90   .jRU...M+....]r.
01c0 - df 69 69 2a 57 60 b0 25-dd 60 fc 93 48 10 96 4d   .ii*W`.%.`..H..M
01d0 - 5d 64 80 4a 77 a7 6a e3-35 ec 31 a2 a6 d4 76 26   ]d.Jw.j.5.1...v&
01e0 - 7e 99 9a 31 7d 41 a0 de-75 b8 a7 00 ae a9 29 9d   ~..1}A..u.....).
01f0 - 47 e0 33 36 8a 63 2b e7-31 c1 8a 5b 05 67 1e af   G.36.c+.1..[.g..
0200 - 3f 1b b3 f0 d9 0b bc e4-a8 5f 05 a8 4d 18 35 0e   ?........_..M.5.
0210 - 06 2d 81 c0 7a a8 16 af-2c cf 80 d5 fb d4 f6 07   .-..z...,.......
0220 - b1 22 8a cb 50 17 0f 64-11 55 1e 08 40 15 5a c4   ."..P..d.U..@.Z.
0230 - a8 2f 37 db b8 39 61 c6-fd 7a f2 51 16 b7 e7 28   ./7..9a..z.Q...(
0240 - 8d f9 1a b6 42 bd e7 b0-88 e5 a1 e8 ec da b3 ed   ....B...........
0250 - 28 1d 04 2e ac 2a 1b 10-a8 18 3c 01 e4 ce cf 68   (....*....<....h
0260 - e1 a6 55 dc b7 ed b8 3a-7b e3 39 fe 6d 9a 1f 5b   ..U....:{.9.m..[
0270 - 4c c2 54 1e 19 37 61 9c-24 b3 6b 22 93 ce f4 cf   L.T..7a.$.k"....
0280 - e5 36 72 d3 f6 42 c7 a9-f5 38 6c 5a 67 e0 ce e0   .6r..B...8lZg...
0290 - 3e 3d f9 48 39 2d 4f 0b-3e 85 e0 81 cd 53 37 f9   >=.H9-O.>....S7.
02a0 - fb c8 55 e4 45 95 91 fc-09 44 3e 6b 23 4c 6e b9   ..U.E....D>k#Ln.
02b0 - b0 02 97 53 f2 35 79 e2-2d fd b6 2d d2 1c 40 b5   ...S.5y.-..-..@.
02c0 - 27 e4 68 eb 67 b0 50 fb-5e 30 a9 0b 23 73 38 c3   '.h.g.P.^0..#s8.
02d0 - a1 c8 2a a7 9f 3e b9 cf-84 63 30 a6 34 17 19 2d   ..*..>...c0.4..-
02e0 - a4 60 65 9b 7c 5f be 2a-ca 71 d0 8d de 9b 4a 88   .`e.|_.*.q....J.
02f0 - 3e 89 c6 09 a6 e3 8d 53-79 e5 de ad 95 d9 c7 e9   >.......Sy.......
0300 - ac db 8f ba 99 98 c8 05-30 8c 4c 48 c6 c5 b7 5d   ........0.LH...]
0310 - 3d a4 81 63 86 5d 25 be-c6 24 e2 d9 1a 6a 0e ba   =..c.]%..$...j..
0320 - cb ee be c8 27 8d 52 57-b7 01 8a ad e1 ba d5 1d   ....'.RW........
0330 - 45 cb 13 3f 78 a4 17 c4-87 12 b9 68 08 7f 4a 12   E..?x......h..J.
0340 - c0 4f 52 35 d5 ae 8e 6c-d0 5e 5b c5 bf 70 dd 2b   .OR5...l.^[..p.+
0350 - 65 7a d9 93 2a 8d 0b 1c-fb be 46 e8 da 39 bb 2b   ez..*.....F..9.+
0360 - 1e 30 e0 a6 30 f3 4b 6d-58 1a 71 62 d6 5c 54 03   .0..0.KmX.qb.\T.
0370 - 7a 2b 01 6f be 11 30 be-45 7a 80 c2 0c 58 93 11   z+.o..0.Ez...X..
0380 - b3 6a 67 7f e0 9a 4b ee-69 df be 31 e3 95 18 f8   .jg...K.i..1....
0390 - f2 91 4b ec 2d 3e b3 d9-68 a2 87 0b 41 fe 27 65   ..K.->..h...A.'e
03a0 - 30 8a 0b ad 7b 1b 3f 28-0e d2 b7 f8 3b ee da a3   0...{.?(....;...
03b0 - 2d 9d ae b1 45 10 a4 92-3e 4e 30 20 50 a6 cb 4a   -...E...>N0 P..J
```

Tabla 3.4 Página WML cifrada.

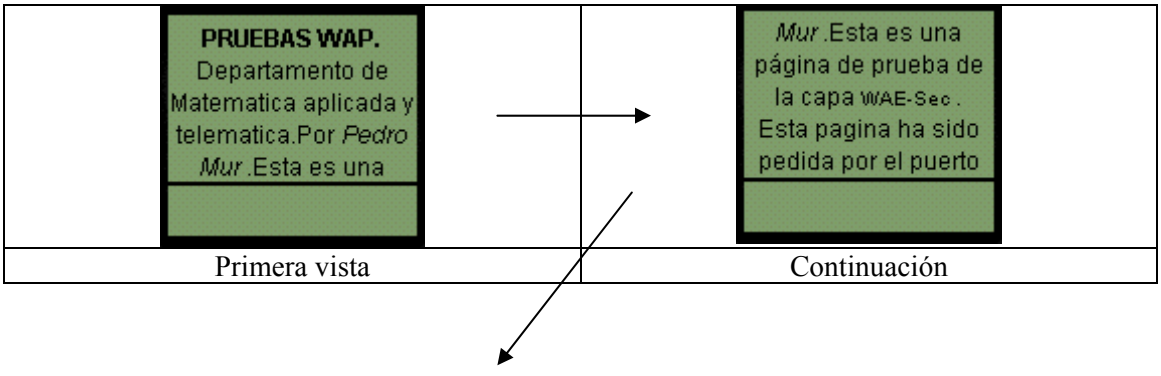
Es decir, que la información que viaja por la red (primero inalámbrica luego cableada) es la mostrada en la tabla 3.4. Dicha figura nos da la información de los datos cifrados a la izquierda en hexadecimal, y a la derecha en ASCII.

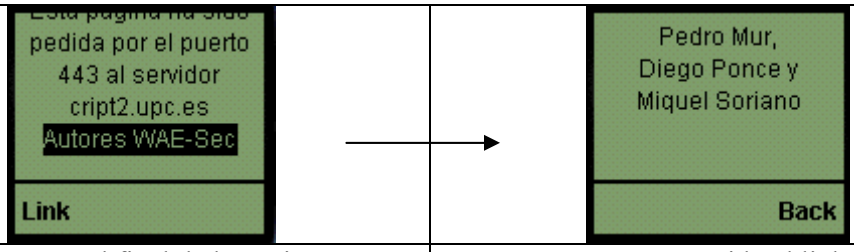
Sin embargo, también podemos enviar la página WML en el formato WBXML, el formato binario de contenidos XML diseñado para reducir el tamaño de los documentos XML transmitidos, permitiendo un uso más efectivo del ancho de banda del canal de comunicación, resulta al pasar el código de la página por un codificador tokenizando las marcas (tags) con el fin de comprimir código. Estos "tokens" son conocidos tanto por emisor como por receptor y debido a que las páginas WML tienen una estructura muy parecida y bastantes elementos en común, WAP Forum pensó en tokenizar los mensajes enviados en WML para así transmitir menos carga sin pérdida de funcionalidad o información semántica. En la tabla 3.5 se puede observar el resultado de la página WBXML mientras que en negrita se pueden ver las descripciones de la primera fila de los tokens de la página.

0000 – **01 01 6ª** 00 **7f e7 55 03**-69 6e 69 74 00 **23 01** e0	..j...U.init.#..
0010 – 07 07 64 03 50 52 55 45-42 41 53 20 57 41 50 2e	..d.PRUEBAS WAP.
0020 – 00 01 26 03 20 44 65 70 61 72 74 61 6d 65 6e 74	..&. Departament
0030 – 6f 20 64 65 20 4d 61 74-65 6d 61 74 69 63 61 20	o de Matemática
0040 – 61 70 6c 69 63 61 64 61-20 79 20 74 65 6c 65 6d	aplicada y telem
0050 – 61 74 69 63 61 2e 50 6f-72 00 6d 03 20 50 65 64	atica.Por.m. Ped
0060 – 72 6f 20 4d 75 72 20 00-01 03 20 2e 45 73 74 61	ro Mur ... .Esta
0070 – 20 65 73 20 75 6e 61 20-70 c3 a1 67 69 6e 61 20	es una p..gina
0080 – 64 65 20 70 72 75 65 62-61 20 64 65 20 6c 61 20	de prueba de la
0090 – 63 61 70 61 20 00 78 03-57 41 45 2d 53 65 63 00	capa .x.WAE-Sec.
00a0 – 01 03 2e 20 00 26 03 20-45 73 74 61 20 70 61 67	... .&. Esta pag
00b0 – 69 6e 61 20 68 61 20 73-69 64 6f 20 70 65 64 69	ina ha sido pedi
00c0 – 64 61 20 70 6f 72 20 65-6c 20 70 75 65 72 74 6f	da por el puerto
00d0 – 20 34 34 33 20 61 6c 20-73 65 72 76 69 64 6f 72	443 al servidor
00e0 – 20 63 72 69 70 74 32 2e-75 70 63 2e 65 73 20 00	cript2.upc.es .
00f0 – 26 dc 4ª 03 23 6d 6f 72-65 00 01 03 41 75 75 6f	&.J.#more...Auto
0100 – 72 65 73 20 57 41 45 2d-53 65 63 00 01 01 01 e7	res WAE-Sec.....
0110 – 55 03 6d 6f 72 65 00 01-e8 46 18 03 62 61 63 6b	U.more...F..back
0120 – 00 01 ab 4ª 03 23 69 6e-69 74 00 01 01 e0 07 01	...J.#init.....
0130 – 03 20 50 65 64 72 6f 20-4d 75 72 2c 00 26 03 20	. Pedro Mur,.&.
0140 – 44 69 65 67 6f 20 50 6f-6e 63 65 20 79 20 00 26	Diego Ponce y .&
0150 – 03 20 4d 69 72 75 65 6c-20 53 6f 72 69 61 6e 6f	. Miquel Soriano
0160 – 20 00 01 01 01	....

Tabla 3.5 Página WBXML en claro.

Se puede observar claramente que la página se reduce de tamaño, a excepción del texto en sí (un carácter en ASCII no puede ser comprimido mediante tokens ya que hay tantos caracteres como combinaciones de los bits que tienen (2^8=256). Y finalmente la página vista desde un teléfono móvil (desde nuestro emulador/visualizador)





Llegamos al final de la página	Hemos seguido el link.

Fig. 3.8 Página vista desde el terminal móvil.

También se han hecho pruebas con archivos de dibujo. Se transmitió un fichero con formato WBMP, obteniendo resultados totalmente satisfactorios y esperados.

3.5.2 Resultados en la pasarela.

En cuanto a la traducción entre HTML y WML (que especifica el protocolo WAP que debe hacer la pasarela WAP), el hecho de haber establecido un túnel seguro entre cliente y servidor hace imposible este hecho. Además la información que viaja en ningún caso sería HTML sino HTML cifrado y por tanto, la pasarela, que no tiene ningún tipo de información sobre el cifrado que se está utilizando nunca podría saber cual es el código HTML que se está transmitiendo.

Las principales funcionalidades de las que se dotó a la pasarela fueron las siguientes:

- Router.
- Traductor de protocolos.
- Capturador de paquetes en las dos direcciones.

Como resultado se obtuvo la tabla 3.6 una vez establecida la comunicación entre cliente y servidor.

Sent 168 rcvd 84 reqst 0 resp 84 unk 0 refused 0		
Active RIP output interfaces		
Destination address	**Interval**	**Split**
192.168.2.1	1	0
192.168.1.1	1	0
192.168.2.2	1	0
192.168.1.225	1	0

Tabla 3.6 del RIP de la pasarela.

Para poder establecer la primera conexión de las máquinas fue necesario especificar a las máquinas las rutas exactas que debían de seguir para comunicarse. Una vez estas rutas fueron creadas, el protocolo de enrutamiento ya se encargaría de actualizarlas como correspondiera en el caso de mover las máquinas.

También existen dos posibilidades de encaminar, el enrutamiento directo y el indirecto. El primero se utiliza cuando los paquetes que hay que traspasar son de una misma máquina, lo que se conoce con el nombre de “forward”, es decir, paso de paquetes. Esta configuración fue necesaria dentro de la máquina que actuaba como pasarela ya que disponía de dos tarjetas de red, una para comunicarse con el cliente y otra para hacer lo propio con el servidor. Por tanto los paquetes procedentes de uno eran traspasados al otro mediante el previo traspaso de un interfaz al otro.

El enrutamiento indirecto es el usado cuando fuente y destino no son la misma máquina y por tanto hay que indicar a los paquetes salientes la ruta que deben seguir para alcanzar destino. Este tipo fue el utilizado en las tablas que debían utilizar cliente y servidor para enviar sus paquetes.

Las tablas de enrutamiento se tuvieron que realizar manualmente mediante la instrucción: route add [ip destino] [gateway] [metric] conformando luego una tabla como la que se puede ver en la tabla 3.7.

Destination	Len	Interface	Gateway	Metric	P	Time	Use
255.255.255.255	32	Eth		1	P	0	0
192.168.2.255	32	Pp0		1	P	0	0
192.168.1.0	24	Pp0	192.168.2.2	7		0	305
147.83.39.0		Eth	192.168.2.2	1		0	0
192.168.2.0		Eth	192.168.2.2	5		0	305
192.168.1.1		Pp0	192.168.2.2	0		0	0

Tabla 3.7 Enrutamiento de la pasarela.

De la tabla 3.7 se desprende que según de donde provengan los paquetes saldrán por la interfaz contraria al que han venido ya que en principio las funciones propias de enrutamiento de la pasarela sólo fueron utilizadas para unir las dos máquinas cliente-servidor. Cada una de las dos direcciones IP de las que dispone la pasarela corresponde a un interfaz de red diferente, de los dos que se configuraron, el de ethernet y el de PPP.

Una vez realizadas todas las aplicaciones, modificaciones y configuraciones se generaron las transmisiones con el servidor siendo éstas totalmente correctas. Se cumplió el objetivo que era que la capa de aplicación WAE pudiera mandar los paquetes a la capa de aplicación del servidor. Estos paquetes salen de la capa WAE quien los transmite para abajo hasta formar los datagramas IP, que con las cabeceras PPP son mandados al servidor pasando eso sí por una pasarela según el modelo WAP.

Ya que se trata de una conexión segura bajo TLS se ha tenido que realizar un handshake con el servidor. Tanto los datos trasmitidos como los recibidos por el cliente correspondientes al handshake en binario y en ASCII se pueden observar en la tabla 3.8.

```
write to 080DBAC8 [080DC050] (130 bytes => 130 (0x82))
0000 - 80 80 01 03 01 00 57 00-00 00 20 00 00 16 00 00   ......W... .....
0010 - 13 00 00 0a 07 00 c0 00-00 66 00 00 07 00 00 05   .........f......
0020 - 00 00 04 05 00 80 03 00-80 01 00 80 08 00 80 00   ................
0030 - 00 65 00 00 64 00 00 63-00 00 62 00 00 61 00 00   .e..d..c..b..a..
0040 - 60 00 00 15 00 00 12 00-00 09 06 00 40 00 00 14   `...........@...
0050 - 00 00 11 00 00 08 00 00-06 00 00 03 04 00 80 02   ................
0060 - 00 80 fd f8 c0 7e 09 1c-55 3e 52 f3 56 4c 95 fe   .....~..U>R.VL..
0070 - fa 9e 48 17 54 91 f1 2f-24 6c ab da bf 9c 99 31   ..H.T../$l.....1
0080 - 83 eb                                             ..
read from 080DBAC8 [080E15B0] (7 bytes => 7 (0x7))
0000 - 16 03 01 00 4a 02                                 ....J.
0007 - <SPACES/NULS>
read from 080DBAC8 [080E15B7] (72 bytes => 72 (0x48))
0000 - 00 46 03 01 39 89 50 95-36 20 5a 55 0a 1f eb a2   .F..9.P.6 ZU....
0010 - 3c 73 5e f3 c8 a0 5e 25-e4 98 c9 94 34 d8 5e 8f   <s^...^%....4.^.
0020 - 66 c6 72 ca 20 45 b6 f7-38 f9 14 78 9d 47 ef ad   f.r. E..8..x.G..
0030 - 8e 1f 5e 9c 3d 61 81 d4-51 27 3b 3b b4 6d df 8b   ..^.=a..Q';;.m..
```

```
0040 - 03 b4 ab 30 36 00 16                              ...06..
0048 - <SPACES/NULS>
read from 080DBAC8 [080E15B0] (5 bytes => 5 (0x5))
0000 - 16 03 01 03 af                                    .....
read from 080DBAC8 [080E15B5] (943 bytes => 940 (0x3AC))
0000 - 0b 00 03 ab 00 03 a8 00-03 a5 30 82 03 a1 30 82   ..........0...0.
0010 - 03 0a a0 03 02 01 02 02-01 01 30 0d 06 09 2a 86   ..........0...*.
0020 - 48 86 f7 0d 01 01 04 05-00 30 81 8a 31 0b 30 09   H........0..1.0.
0030 - 06 03 55 04 06 13 02 45-53 31 12 30 10 06 03 55   ..U....ES1.0...U
0040 - 04 08 13 09 43 41 54 41-4c 55 4e 59 41 31 12 30   ....CATALUNYA1.0
0050 - 10 06 03 55 04 07 13 09-42 41 52 43 45 4c 4f 4e   ...U....BARCELON
0060 - 41 31 0c 30 0a 06 03 55-04 0a 13 03 55 50 43 31   A1.0...U....UPC1
0070 - 0e 30 0c 06 03 55 04 0b-13 05 4d 41 69 54 45 31   .0...U....MAiTE1
0080 - 15 30 13 06 03 55 04 03-13 0c 41 75 74 6f 72 69   .0...U....Autori
0090 - 64 61 64 20 49 49 31 1e-30 1c 06 09 2a 86 48 86   dad II1.0...*.H.
00a0 - f7 0d 01 09 01 16 0f 70-6d 75 72 40 6d 61 74 2e   .......pmur@mat.
00b0 - 75 70 63 2e 65 73 30 1e-17 0d 30 30 30 35 32 39   upc.es0...000529
00c0 - 31 32 35 39 31 31 5a 17-0d 30 31 30 35 32 39 31   125911Z..0105291
00d0 - 32 35 39 31 31 5a 30 81-87 31 0b 30 09 06 03 55   25911Z0..1.0...U
00e0 - 04 06 13 02 45 53 31 12-30 10 06 03 55 04 08 13   ....ES1.0...U...
00f0 - 09 43 41 54 41 4c 55 4e-59 41 31 0c 30 0a 06 03   .CATALUNYA1.0...
0100 - 55 04 07 13 03 42 43 4e-31 0e 30 0c 06 03 55 04   U....BCN1.0...U.
0110 - 0a 13 05 4d 41 69 54 45-31 0e 30 0c 06 03 55 04   ...MAiTE1.0...U.
0120 - 0b 13 05 4d 41 69 54 45-31 16 30 14 06 03 55 04   ...MAiTE1.0...U.
0130 - 03 13 0d 63 72 69 70 74-31 2e 75 70 63 2e 65 73   ...cript1.upc.es
0140 - 31 1e 30 1c 06 09 2a 86-48 86 f7 0d 01 09 01 16   1.0...*.H.......
0150 - 0f 70 6d 75 72 40 6d 61-74 2e 75 70 63 2e 65 73   .pmur@mat.upc.es
0160 - 30 81 9f 30 0d 06 09 2a-86 48 86 f7 0d 01 01 01   0..0...*.H......
0170 - 05 00 03 81 8d 00 30 81-89 02 81 81 00 c8 86 a2   ......0.........
0180 - 4e c7 80 3d c5 2d 8a 73-c0 f4 12 81 fb 47 4e f6   N..=.-.s.....GN.
0190 - 97 d9 3a 95 98 42 d8 89-2d fb c4 94 50 9d 54 0c   ..:..B..-...P.T.
01a0 - 37 db e4 7a 1b 6f f3 00-88 9b a1 7e bc 46 83 ac   7..z.o.....~.F..
01b0 - 33 1f 8e a3 2e 7d ac 92-cf 8f eb 9b 22 5c f1 81   3....}......"\..
01c0 - 92 17 76 7e 5a d2 47 d6-0b f3 d0 f5 0f ab 08 a5   ..v~Z.G.........
01d0 - fc e0 35 b3 a9 72 e2 8d-91 04 48 8b 3d 15 14 89   ..5..r....H.=...
01e0 - 45 97 79 44 ad 53 a8 00-ee 62 bb cf bd b8 b3 cd   E.yD.S...b......
01f0 - 23 09 47 ea 1a f9 1f aa-18 29 ec 25 77 02 03 01   #.G......).%w...
0200 - 00 01 a3 82 01 16 30 82-01 12 30 09 06 03 55 1d   ......0...0...U.
0210 - 13 04 02 30 00 30 2c 06-09 60 86 48 01 86 f8 42   ...0.0,..`.H...B
0220 - 01 0d 04 1f 16 1d 4f 70-65 6e 53 53 4c 20 47 65   ......OpenSSL Ge
0230 - 6e 65 72 61 74 65 64 20-43 65 72 74 69 66 69 63   nerated Certific
0240 - 61 74 65 30 1d 06 03 55-1d 0e 04 16 04 14 79 95   ate0...U......y.
0250 - 74 27 99 e9 d8 e3 69 ce-5a ea d9 a3 54 2d d3 ea   t'....i.Z...T-..
0260 - be 9d 30 81 b7 06 03 55-1d 23 04 81 af 30 81 ac   ..0....U.#...0..
0270 - 80 14 ac 10 fb 89 a7 6a-44 cc e4 32 60 09 9a 58   .......jD..2`..X
0280 - 91 02 d2 d2 f6 31 a1 81-90 a4 81 8d 30 81 8a 31   .....1......0..1
0290 - 0b 30 09 06 03 55 04 06-13 02 45 53 31 12 30 10   .0...U....ES1.0.
02a0 - 06 03 55 04 08 13 09 43-41 54 41 4c 55 4e 59 41   ..U....CATALUNYA
02b0 - 31 12 30 10 06 03 55 04-07 13 09 42 41 52 43 45   1.0...U....BARCE
02c0 - 4c 4f 4e 41 31 0c 30 0a-06 03 55 04 0a 13 03 55   LONA1.0...U....U
02d0 - 50 43 31 0e 30 0c 06 03-55 04 0b 13 05 4d 41 69   PC1.0...U....MAi
02e0 - 54 45 31 15 30 13 06 03-55 04 03 13 0c 41 75 74   TE1.0...U....Aut
02f0 - 6f 72 69 64 61 64 20 49-49 31 1e 30 1c 06 09 2a   oridad II1.0...*
0300 - 86 48 86 f7 0d 01 09 01-16 0f 70 6d 75 72 40 6d   .H........pmur@m
0310 - 61 74 2e 75 70 63 2e 65-73 82 01 00 30 0d 06 09   at.upc.es...0...
0320 - 2a 86 48 86 f7 0d 01 01-04 05 00 03 81 81 00 48   *.H............H
0330 - a3 34 98 56 04 5e 3f 80-48 07 d3 c1 d3 8c 0a 8d   .4.V.^?.H.......
0340 - 59 70 0a 34 4e 4e a2 4a-a4 8b 15 0e 1e c8 d5 5f   Yp.4NN.J.......
0350 - bb 38 3d 1b d5 cb e9 4c-1c d6 2d 49 f6 83 8b a7   .8=....L..-I....
0360 - 78 a4 c8 69 cf c5 f1 78-97 b9 a6 79 bf a3 1a 2c   x..i...x...y...,
0370 - bb 4e bd 3c ce 6c d2 d1-65 a8 a9 94 f1 6d 02 17   .N.<.l..e....m..
0380 - a6 44 c5 47 88 29 83 de-8c f0 6c 23 69 38 72 2e   .D.G.)....l#i8r.
0390 - 5d e6 70 c2 d7 8e e7 61-dc e8 3b e9 7e b4 ce da   ].p....a..;.~...
```

```
03a0 - 41 5d 81 cf d5 18 b5 34-80 46 ce e8               A].....4.F..
read from 080DBAC8 [080E1961] (3 bytes => 3 (0x3))
0000 - 73 e1 a7                                          s..
read from 080DBAC8 [080E15B0] (5 bytes => 5 (0x5))
0000 - 16 03 01 01 8d                                    .....
read from 080DBAC8 [080E15B5] (397 bytes => 397 (0x18D))
0000 - 0c 00 01 89 00 80 e6 96-9d 3d 49 5b e3 2c 7c f1   .........=I[.,|.
0010 - 80 c3 bd d4 79 8e 91 b7-81 82 51 bb 05 5e 2a 20   ....y.....Q..^*
0020 - 64 90 4a 79 a7 70 fa 15-a2 59 cb d5 23 a6 a6 ef   d.Jy.p...Y..#...
0030 - 09 c4 30 48 d5 a2 2f 97-1f 3c 20 12 9b 48 00 0e   ..0H../..< ..H..
0040 - 6e dd 06 1c bc 05 3e 37-1d 79 4e 53 27 df 61 1e   n.....>7.yNS'.a.
0050 - bb be 1b ac 9b 5c 60 44-cf 02 3d 76 e0 5e ea 9b   .....\`D..=v.^..
0060 - ad 99 1b 13 a6 3c 97 4e-9e f1 83 9e b5 db 12 51   .....<.N.......Q
0070 - 36 f7 26 2e 56 a8 87 15-38 df d8 23 c6 50 50 85   6.&.V...8..#.PP.
0080 - e2 1f 0d d5 c8 6b 00 01-02 00 80 b4 1a cc 1f 0e   .....k..........
0090 - ac fe 6d 69 6b 32 32 dc-d8 87 a2 b5 50 1b 24 bd   ..mik22.....P.$.
00a0 - 32 a0 95 b7 b5 c2 37 a5-1d c7 6a dc 6d c1 d9 76   2.....7...j.m..v
00b0 - cf 76 8c 0f b2 13 8d 4f-c9 81 c2 55 98 cd 6c 25   .v.....O...U..l%
00c0 - 0d 74 dd eb f2 ba 53 4f-19 75 de 41 08 86 7a 2f   .t....SO.u.A..z/
00d0 - 1d 19 fd ab 86 b4 54 af-d6 19 7b e6 f7 31 dd 5a   ......T...{..1.Z
00e0 - ec 1b 7c 5f 44 01 03 ba-d7 78 21 1a 98 ac cd 72   ..|_D....x!....r
00f0 - be 1f 91 8a 90 87 8a 04-f4 2d 2a 69 fa 4c 85 6d   .........-*i.L.m
0100 - ab f9 f0 cc 96 e5 ae 31-f6 30 1d 00 80 00 1d ca   .......1.0......
0110 - 6e d2 38 56 3d 0b d4 fa-80 37 b4 b4 18 08 30 b6   n.8V=....7....0.
0120 - 9c 77 13 d1 4c 28 23 24-99 d5 41 d8 30 27 d7 89   .w..L(#$..A.0'..
0130 - bd cb 76 2d 96 df dc c4-3b be 49 cf 9f 61 80 14   ..v-....;.I..a..
0140 - b5 b7 16 39 91 d7 97 30-2f 95 6b 2d 7f 67 a9 06   ...9...0/.k-.g..
0150 - 6f 37 6e 00 7b c1 af cc-eb a0 d1 7c 56 df 26 4f   o7n.{......|V.&O
0160 - 69 61 1c 35 d4 85 ed c4-f9 43 b7 2c a9 eb 84 91   ia.5.....C.,....
0170 - 42 99 9d 67 0a dd e8 2f-cb 8d 99 63 4e 58 05 40   B..g.../...cNX.@
0180 - 71 b9 d3 69 67 b1 3a 8c-cd 35 48 64 98            q..ig.:..5Hd.
read from 080DBAC8 [080E15B0] (5 bytes => 5 (0x5))
0000 - 16 03 01 00 04                                    .....
read from 080DBAC8 [080E15B5] (4 bytes => 4 (0x4))
0000 - 0e                                                .
0004 - <SPACES/NULS>
write to 080DBAC8 [080EAA48] (139 bytes => 139 (0x8B))
0000 - 16 03 01 00 86 10 00 00-82 00 80 e4 ab ed 89 cc   ................
0010 - 85 60 4c a5 75 ab a6 f9-eb 3e e6 a9 a7 c1 b4 f1   .`L.u....>......
0020 - 60 ab bf a1 56 ee a3 7d-d7 62 3e f7 ed 22 c6 97   `...V..}.b>.."..
0030 - aa e6 f7 b3 c4 92 71 aa-ad 0c e1 34 f7 8d 88 70   ......q....4...p
0040 - 57 c0 27 4b a2 d0 53 04-35 44 54 df 67 c5 7f 7d   W.'K..S.5DT.g..}
0050 - ac 21 b7 c7 38 e3 d7 ea-e3 7b b1 de ce 7f 1d b8   .!..8....{......
0060 - 23 c4 40 95 44 fa 1b 71-ba 92 61 fb e3 e0 b3 ba   #.@.D..q..a.....
0070 - ef e2 11 8f 5f 57 72 84-46 32 9a 42 aa 3e 15 3b   ...._Wr.F2.B.>.;
0080 - 4e 9d 20 5f f6 80 e7 a0-e5 f4 9d                  N. _........
write to 080DBAC8 [080EAA48] (6 bytes => 6 (0x6))
0000 - 14 03 01 00 01 01                                 ......
write to 080DBAC8 [080EAA48] (45 bytes => 45 (0x2D))
0000 - 16 03 01 00 28 4f 67 8a-c7 de 03 f4 53 50 ac c1   ....(Og.....SP..
0010 - 10 e8 f2 a1 f1 38 c8 67-c2 d0 c4 7e fb e8 e5 ba   .....8.g...~....
0020 - 8b f3 92 7b ca f7 9f c1-a3 64 ec 51 0d            ...{.....d.Q.
read from 080DBAC8 [080E15B0] (5 bytes => 5 (0x5))
0000 - 14 03 01 00 01                                    .....
read from 080DBAC8 [080E15B5] (1 bytes => 1 (0x1))
0000 - 01                                                .
read from 080DBAC8 [080E15B0] (5 bytes => 5 (0x5))
0000 - 16 03 01 00 28                                    ....(
read from 080DBAC8 [080E15B5] (40 bytes => 40 (0x28))
0000 - 3f b8 c5 2e 41 63 11 ed-bb a9 84 04 e6 15 d0 84   ?...Ac..........
0010 - 63 f9 4c bf ee 6b 06 7d-fd bf 2b 7f 43 0b 37 ca   c.L..k.}..+.C.7.
0020 - 0e c9 3f eb 95 7d 86 7e-                          ..?..}.~
```

Tabla 3.8 Datos del handshake.

Los primeros 130 bytes enviados por el cliente corresponden al hello del cliente para la versión del protocolo TLSv1. Seguidamente recibe el *hello* del servidor, que está compuesto por el hola del servidor, el certificado y la clave del servidor (7+72+5+940+3+5+397 bytes) y el mensaje de hola de servidor efectuado (4 bytes). Debido a que el servidor no exige la autentificación del cliente, éste le responde únicamente con el intercambio de clave de cliente (139+5 bytes) acabando con un mensaje de finalizado (51 bytes). El servidor lo recibe y le devuelve un mensaje de finalizado (51 bytes).

Una vez realizado el handshake, tanto cliente como servidor ya se han puesto de acuerdo en como se va a cifrar la conexión y que seguridad se le va a aplicar (claves, certificados, etc.). Estos datos se pueden ver en la tabla 3.9.

```
Certificate chain
 0 s:/C=ES/ST=CATALUNYA/L=BCN/O=MAiTE/OU=MAiTE/CN=cript1.upc.es/Email=pmur@mat.upc.es
   i:/C=ES/ST=CATALUNYA/L=BARCELONA/O=UPC/OU=MAiTE/CN=Autoridad II/Email=pmur@mat.upc.es
Server certificate
-----BEGIN CERTIFICATE-----
MIIDoTCCAwqgAwIBAgIBATANBgkqhkiG9w0BAQQFADCBijELMAkGA1UEBhMCRVMx
EjAQBgNVBAgTCUNBVEFMVU5ZQTESMBAGA1UEBxMJQkFSQ0VMT05BMQwwCgYDVQQK
EwNVUEMxDjAMBgNVBAsTBU1BaVRFMRUwEwYDVQQDEwxBdXRvcmlkYWQgSUkxHjAc
BgkqhkiG9w0BCQEWD3BtdXJAbWF0LnVwYy5lczAeFw0wMDA1MjkxMjU5MTFaFw0w
MTA1MjkxMjU5MTFaMIGHMQswCQYDVQQGEwJFUzESMBAGA1UECBMJQ0FUQUxVTllB
MQwwCgYDVQQHEwNCQ04xDjAMBgNVBAoTBU1BaVRFMQ4wDAYDVQQLEwVNQWlURTEW
MBQGA1UEAxMNY3JpcHQxLnVwYy5lczEeMBwGCSqGSIb3DQEJARYPcG11ckBtYXQu
dXBjLmVzMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDIhqJOx4A9xS2Kc8D0
EoH7R072l9k6lZhC2Ikt+8SUUJ1UDDfb5Hobb/MAiJuhfrxGg6wzH46jLn2sks+P
65siXPGBkhd2flrSR9YL89D1D6sIpfzgNbOpcuKNkQRIiz0VFIlFl3lErVOoAO5i
u8+9uLPNIwlH6hr5H6oYKewldwIDAQABo4IBFjCCARIwCQYDVR0TBAIwADAsBglg
hkgBhvhCAQ0EHxYdT3BlblNTTCBHZW5lcmF0ZWQgQ2VydGlmaWNhdGUwHQYDVR0O
BBYEFHmVdCeZ6djjac5a6tmjVC3T6r6dMIG3BgNVHSMEga8wgayAFKwQ+4mnakTM
5DJgCZpYkQLS0vYxoYGQpIGNMIGKMQswCQYDVQQGEwJFUzESMBAGA1UECBMJQ0FU
QUxVTllBMRIwEAYDVQQHEwlCQVJDRUxPTkExDDAKBgNVBAoTA1VQQzEOMAwGA1UE
CxMFTUFpVEUxFTATBgNVBAMTDEF1dG9yaWRhZCBJSTEeMBwGCSqGSIb3DQEJARYP
cG11ckBtYXQudXBjLmVzggEAMA0GCSqGSIb3DQEBBAUAA4GBAEijNJhWBF4/gEgH
08HTjAqNWXAKNE5OokqkixUOHsjVX7s4PRvVy+lMHNYtSfaDi6d4pMhpz8XxeJe5
pnm/oxosu069PM5s0tFlqKmU8W0CF6ZExUeIKYPejPBsI2k4ci5d5nDC147nYdzo
O+l+tM7aQV2Bz9UYtTSARs7oc+Gn
-----END CERTIFICATE-----
subject=/C=ES/ST=CATALUNYA/L=BCN/O=MAiTE/OU=MAiTE/CN=cript1.upc.es/Email=pmur@mat.upc.es
issuer=/C=ES/ST=CATALUNYA/L=BARCELONA/O=UPC/OU=MAiTE/CN=Autoridad II/Email=pmur@mat.upc.es
---
No client certificate CA names sent
---
TLS handshake has read 1489 bytes and written 320 bytes
---
New, TLSv1/TLSv3, Cipher is EDH-RSA-DES-CBC3-SHA
Server public key is 1024 bit
TLS-Session:
    Protocol  : TLSv1
    Cipher    : EDH-RSA-DES-CBC3-SHA
    Session-ID: 45B6F738F914789D47EFAD8E1F5E9C3D6181D451273B3BB46DDF8B03B4AB3036
    Session-ID-ctx:
    Master-Key:
AA3DE2CD41AD78EDC0D8F7822CDE1A1EB599B5E14AC81727288642B273CBC6566933E5BF01C8066D2B3
C063FDB4E36A3
    Key-Arg   : None
    Start Time: 954165921
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
```

Tabla 3.9 Datos criptográficos de la conexión.

Es decir, para esta comunicación se ha negociado una seguridad TLSv1/SSLv3, con cifrado EDH-RSA-DES-CBC3-SHA además de hacerse con los identificadores y claves de la tabla 3.9.

Captura de paquetes en la pasarela.

Otra de las cualidades de las que se dotó a la pasarela construida fue la de capturadora de paquetes con el fin de ver toda la información que pasaba por ella, que en definitiva viene a ser toda la información que está viajando entre la comunicación de cliente y servidor.

Cabe decir que el sistema funcionó a la perfección y cliente y servidor pudieron comunicarse como si de un teléfono móvil y un servidor WAP se tratara. Los errores de la transmisión fueron nulos tal y como se preveía y la velocidad fue la esperada y la aceptada por el cable y en entorno real por GSM, 9600 bps.

Como demostración de que así fue se capturaron los paquetes correspondientes la comunicación WAP segura entre el cliente y el servidor desde el inicio de la conexión, pasando por la negociación del protocolo de seguridad hasta llegar a realizar la petición de la página y ser dada por el servidor.

En la tabla 3.10 se pueden observar los paquetes que vienen del lado del cliente, es decir, los que llegan a la pasarela mediante el protocolo punto a punto PPP.

```
Wed Aug 09 13:47:26 2000 - pp0 recv:
PPP: len  48          protocol: IP
IP: len 44 192.168.1.225->192.168.2.2 ihl 20 ttl 254 prot TCP

Wed Aug 09 13:47:26 2000 - pp0 recv:
PPP: len  44          protocol: VJ Uncompressed TCP/IP
          connection 0x00
IP: len 40 192.168.1.225->192.168.2.2 ihl 20 ttl 254 prot TCP
0000  E..(.....7.............`..t..P.....

Wed Aug 09 13:47:28 2000 - pp0 recv:
PPP: len 137          protocol: VJ Compressed TCP/IP
          changes: 0x10   TCP checksum: 0xc4b0   PUSH
          increment ID
0000  .....W... .............f.....................e..d..c..b..a..
0040  `...........@.....................m.Sh...Q...w.uZ.ihx.DS....=R..
0080  J

Wed Aug 09 13:47:30 2000 - pp0 recv:
PPP: len  11          protocol: VJ Compressed TCP/IP
          changes: 0x0c   TCP checksum: 0xfe1f
          delta ACK: 0x400   delta SEQ: 0x82     increment ID

Wed Aug 09 13:47:30 2000 - pp0 recv:
PPP: len  10          protocol: VJ Compressed TCP/IP
          changes: 0x04   TCP checksum: 0xfc7d
          delta ACK: 0x1a2   increment ID

Wed Aug 09 13:47:31 2000 - pp0 recv:
PPP: len 197          protocol: VJ Compressed TCP/IP
          changes: 0x10   TCP checksum: 0x1c25   PUSH
          increment ID
0000  .............2g..!...Q.y..'ro....._...e.c..../.S.O.........0....
0040  %.]~.m..>1.\."....5..S.....L....8....87..M.e...%...].b.._.'....8
0080  .t.2_..oa.............(]O?.n.<.7.][..}...7.K.....R7.e....D!..w{
Wed Aug 09 13:47:31 2000 - pp0 recv:
PPP: len   9          protocol: VJ Compressed TCP/IP
```

```
            changes: 0x0c   TCP checksum: 0xfb8c
            delta ACK: 0x33     delta SEQ: 0xbe      increment ID

Wed Aug 09 13:47:34 2000 - pp0 recv:
PPP: len 118            protocol: VJ Compressed TCP/IP
            changes: 0x10   TCP checksum: 0x039c   PUSH
            increment ID
0000  ....0].5.....9.....!.J.....K.z......D.......>.F.....W,.&R.
0040  ....,.......t..u..... ...Z.z....!.

Wed Aug 09 13:47:35 2000 - pp0 recv:
PPP: len  11            protocol: VJ Compressed TCP/IP
            changes: 0x0c   TCP checksum: 0xf758
            delta ACK: 0x3c5   delta SEQ: 0x6f      increment ID

Wed Aug 09 13:47:35 2000 - pp0 recv:
PPP: len   8            protocol: VJ Compressed TCP/IP
            changes: 0x04   TCP checksum: 0xf73b
            delta ACK: 0x1d     increment ID

Wed Aug 09 13:47:35 2000 - pp0 recv:
PPP: len   8            protocol: VJ Compressed TCP/IP
            changes: 0x04   TCP checksum: 0xf73a
            delta ACK: 0x01     increment ID

Wed Aug 09 13:47:38 2000 - pp0 recv:
PPP: len  44            protocol: IP
IP: len 40 192.168.1.225->192.168.2.2 ihl 20 ttl 254 prot TCP
```

Tabla 3.10 Captura de paquetes dirección cliente → servidor por PPP.

En la tabla 3.11 se pueden observar los paquetes que vienen del lado del servidor, los que llegan por la tarjeta ethernet. Se pueden ver claramente los datos significativos que están circulando como pueden ser el certificado del servidor y sobretodo la página WML pedida, que es fin de cuentas los que se persigue. Esta página cifrada es la que viaja por la red y la que podría capturar cualquier *hacker o ladrón de la red*. Se puede apreciar claramente que la información es del todo incomprensible si se desconocen los datos criptográficos con los que ha sido creada.

Se han marcado en negrita el certificado enviado así como la página cifrada, llegando a la conclusión de que una persona que no sea ni emisor ni receptor, en principio, no puede ver cual es la página en claro.

```
Wed Aug 09 13:42:13 2000 - eth recv:
Ether: len 60 00:60:97:16:88:cb->00:20:af:5c:93:3c type IP
IP: len 44 192.168.2.2->192.168.1.225 ihl 20 ttl 64 DF prot TCP
TCP: 443->1024 Seq x8d65477a Ack x742c001 ACK SYN Wnd 31744 MSS 1024

Wed Aug 09 13:42:15 2000 - eth recv:
Ether: len 60 00:60:97:16:88:cb->00:20:af:5c:93:3c type IP
IP: len 40 192.168.2.2->192.168.1.225 ihl 20 ttl 64 DF prot TCP
TCP: 443->1024 Seq x8d65477b Ack x742c083 ACK Wnd 31744

Wed Aug 09 13:42:15 2000 - eth recv:
Ether: len 1078 00:60:97:16:88:cb->00:20:af:5c:93:3c type IP
IP: len 1064 192.168.2.2->192.168.1.225 ihl 20 ttl 64 DF prot TCP
TCP: 443->1024 Seq x8d65477b Ack x742c083 ACK PSH Wnd 31744 Data 1024
0000  ....J...F..9.B.4.N..v..(iB...*H..i...=.....  ......u/..>.+.....YC
0040  ....M.q..8.....................0...0...........0...*.H........0..
0080  1.0...U....ES1.0...U....CATALUNYA1.0...U....BARCELONA1.0...U....
00c0  UPC1.0...U....MAiTE1.0...U....Autoridad II1.0...*.H........pmur@
```

```
0100  mat.upc.es0...000719152440Z..010719152440Z0..1.0...U....ES1.0...
0140  U....Catalunya1.0...U....Barcelona1.0...U....Maite1.0...U....UPC
0180  1.0...U....cript2.upc.es1.0...*.H........pmur@mat.upc.es0..0...*
01c0  .H.............0.......h....8IR.y..!......iHG./.{....2..5$_..U(;.
0200  ...fT....wz.t..LX..M/.;.u[m.}O.6.. .Yc.u..\...Mx........0C....2.
0240  ...O.(q.Hf....j7e..+7.........0...0...U....0.0,..`.H...B......Op
0280  enTLS Generated Certificate0...U......f....g^.H.....(..'.D0....U
02c0  .#...0.........jD..2`..X.....1......0..1.0...U....ES1.0...U....C
0300  ATALUNYA1.0...U....BARCELONA1.0...U....UPC1.0...U....MAiTE1.0...
0340  U....Autoridad II1.0...*.H........pmur@mat.upc.es...0...*.H.....
0380  .........9....].1..e..m.D%.yM%....P...E.a%v.r.b..krT#{B....K....
03c0  2.....w.9..}K.\....a..............c...K..a.,.O.....c...........

Wed Aug 09 13:42:15 2000 - eth recv:
Ether: len 472 00:60:97:16:88:cb->00:20:af:5c:93:3c type IP
IP: len 458 192.168.2.2->192.168.1.225 ihl 20 ttl 64 DF prot TCP
TCP: 443->1024 Seq x8d654b7b Ack x742c083 ACK PSH Wnd 31744 Data 418
0000  ~..9.................=I[.,|.....y.....Q..^* d.Jy.p...Y..#.....0H
0040  ../..< ..H..n.....>7.yNS'.a......\`D..=v.^.......<.N.......Q6.&.
0080  V...8..#.PP......k......qL.1%[......UF'..n...Q.=.Z.4~.*y].4..c.{
00c0  /...dMiLJ.._cy$o@hT<..|.Y..K..W(j.....8I.a.....\IX.6.............
0100  ...&.........h7..*..(*F.... ......)~.....b.V.[..jU.;....VRAI...M
0140  f.u1NI.S*...K>2...#........_.;..h...I..w..u&C..S....W.4O.},./...
0180  ........y.c...T..w......'.........

Wed Aug 09 13:42:18 2000 - eth recv:
Ether: len 60 00:60:97:16:88:cb->00:20:af:5c:93:3c type IP
IP: len 40 192.168.2.2->192.168.1.225 ihl 20 ttl 64 DF prot TCP
TCP: 443->1024 Seq x8d654d1d Ack x742c141 ACK Wnd 31744

Wed Aug 09 13:42:18 2000 - eth recv:
Ether: len 105 00:60:97:16:88:cb->00:20:af:5c:93:3c type IP
IP: len 91 192.168.2.2->192.168.1.225 ihl 20 ttl 64 DF prot TCP
TCP: 443->1024 Seq x8d654d1d Ack x742c141 ACK PSH Wnd 31744 Data 51
0000  ..........(..E.^.Q_@..a:.ye5k...........9.t..F..D..

Wed Aug 09 13:42:20 2000 - eth recv:
Ether: len 1019 00:60:97:16:88:cb->00:20:af:5c:93:3c type IP
IP: len 1005 192.168.2.2->192.168.1.225 ihl 20 ttl 64 DF prot TCP
TCP: 443->1024 Seq x8d654d50 Ack x742c1b0 ACK PSH Wnd 31744 Data 965
0000  ......R...2....fE.'.6ql.[.wH".@._'.f..`.w5.bV...D.]Z2'Z........?
0040  ..gC.........z..... .,.Ih..M...2$As......EcM&!.....G.Y.hEFKb-..
0080  M...S.MK....P^U8v......4......h76PcJG.B0.q..GY..ns....(.Qd%.m..V
00c0  4k5aq...Q)....)3......L]..X..B../g.2....%.r....Yn...`.. "zxL:.z.
0100  w.S.|a.t..-a...Zr.)w../U....Q..\N!..:......g.....>&:.Og.RXb....3
0140  ...>$.5C.pE....lq...F...p...#.......C.u....BB.3\+%.../...[.[....
0180  .5b.a-v........}.e.....:>T.v.g....d>.C.>...=.s"...L..O.f.....D.N
01c0  ....W...<.L......+..OY....-=......*..,....Fc..9.*..'|c......i.I.
0200  .........b6Q.f.Z..*>.V..........A..........c.v..~.B.d...U^...v.n
0240  c..~....b.V.J..l.A6.......1...,........?.`..5..../.)L4.L.1.B.WJ..
0280  ...S&...)..|.#.........{..B..E.{OD].F..nz.9.B..+.*.3]R,=)PQ@Fl./
02c0  .d.0...1k...PR...3y.iU..w.W....>.\a.<....<..]P..<.FW.C(*-..../w.
0300  ....C.?x...e..T..<..k.+y.{Z...G..w.....dEh..s#e|*........7+.2&R.
0340  =...........N\$4.O...B.3.d..V.......`..FX{..?S.....r....g......P
0380  `...22..T...4....N.UokQe.I..jZ.....X[u.c.z.j[0...e...+\..W..~...
03c0  k....

Wed Aug 09 13:42:20 2000 - eth recv:
Ether: len 83 00:60:97:16:88:cb->00:20:af:5c:93:3c type IP
IP: len 69 192.168.2.2->192.168.1.225 ihl 20 ttl 64 DF prot TCP
TCP: 443->1024 Seq x8d655115 Ack x742c1b0 ACK PSH Wnd 31744 Data 29
0000  .....L2.%.s:.D0E...@..`?.j.O.
```

```
Wed Aug 09 13:42:20 2000 - eth recv:
Ether: len 60 00:60:97:16:88:cb->00:20:af:5c:93:3c type IP
IP: len 40 192.168.2.2->192.168.1.225 ihl 20 ttl 64 DF prot TCP
TCP: 443->1024 Seq x8d655132 Ack x742c1b0 ACK FIN Wnd 31744

Wed Aug 09 13:42:22 2000 - eth recv:
Ether: len 60 00:60:97:16:88:cb->00:20:af:5c:93:3c type IP
IP: len 40 192.168.2.2->192.168.1.225 ihl 20 ttl 64 DF prot TCP
TCP: 443->1024 Seq x8d655133 Ack x742c1b1 ACK Wnd 31744
```

Tabla 3.11 Captura de paquetes dirección servidor → cliente por Ethernet.

3.6 Comparación de las cargas.

Otra de las facilidades de las que se dotó al cliente WAE fue la de poder medir la cantidad de bytes enviados y recibidos en todo momento. Esto permitió sacar resultados y estadísticas sobre la carga útil que supone cifrar ficheros compatibles con el protocolo WAP mediante TLS o mediante WTLS, además de texto en claro.

En la tabla 3.12 se puede observar la carga (siempre en la capa WAE, ya que las capas inferiores que se utilizan dentro del sistema realizado son exactamente las mismas que en la realidad según el modelo WAP y por tanto no aportan ningún cambio) que supone el envío de 3 tipos de ficheros que serían soportados por WAP además de 1 que no lo soporta pero sería el equivalente en Web. En la tabla 3.13 se puede observar la carga que supone el handshake para el cliente tanto en emisión como en recepción, con la relación del tiempo que supondría en distintos ambientes y velocidades.

Fichero	En claro (bytes)	Con TLS (bytes)	Aumento de carga (%)
Index.wml	657	960	46.12%
Index.wmlc	357	656	83.36%
upc.gif	7170	7467	4.14 %
upc.wpmb	1700	2000	15 %

Tabla 3.12 Carga en la capa WAE.

Velocidad de conexión	Bytes leidos	Bytes escritos	Tiempo estimado perdido
9600 bps	1489	320	1.51 seg
1 Kbps	1489	320	14.13 seg
0.1 Kbps	1489	320	2 min 21 seg

Tabla 3.13 Carga del handshake.

Se dispone de un fichero WML, es decir, el fichero compatible con WAP con un tamaño de 657 bytes. Según el estándar WAP, este fichero pasaría por el proceso de tokenización y compresión como se ha explicado anteriormente. Quedaría en un fichero tokenizado de 357 bytes. Es decir, en una transmisión normal y sin seguridad aportaría una carga útil de estos 357 bytes. Esta no es la carga que viajaría por el aire ya que corresponde únicamente a la capa de aplicación (la más alta) de la torre de protocolos WAP. Esta carga iría bajando por todas las capas que pondrían sus cabeceras, partirían en trozos más pequeños y enviarían dependiendo de la portadora utilizada.

En caso de ser una transmisión segura con WTLS, utilizando por ejemplo el algoritmo de seguridad RSA (ya aceptado en la versión 1.2 del estándar WAP [Wap 99], la carga enviada sería de 656 bytes, es decir, habría un aumento de un 83.4 % de la carga, prácticamente se

dobla. Sin embargo, un cifrado sobre un texto predecible es débil y puede ser descubierto por un criptoanalista. Por tanto esta opción, queda descartada.

Pasando ahora a la seguridad propuesta en este proyecto, utilizando TLS sobre el fichero WML directamente se consigue una carga de 960 bytes, 168.91%. Es decir, casi se triplican los datos enviados, a cambio de ofrecer seguridad. En el dominio temporal, se está triplicando el tiempo de descarga de la página pedida, y por consiguiente se está pagando más ya que la duración de la conexión es mayor. Este es el compromiso que se tiene entre seguridad y tiempo/precio. Si solo se tuviera que transmitir este fichero, y a una velocidad de 9600 bps como es capaz de ofrecer actualmente GSM, la portadora más utilizada hoy en día por los terminales WAP, supondría un aumento de 0.3 a 0.8 segundos, prácticamente despreciable por la poca longitud de la página a transmitir.

Sin embargo, son los gráficos en WAP y todos los elementos multimedia (gráficos, videos, animaciones, applets de java) en TCP/IP Web los que generan mayor tráfico en la red y los que suponen mayores retardos en la red. En el caso de WAP, el estándar de dibujos adoptados son el WBMP. En el ejemplo, la carga de este fichero es de 1700 bytes, que aplicándole la capa WAE-Sec propuesta se queda en 2000 bytes a transmitir. Es decir, un aumento de carga del 15 %. Como se puede ver, esta carga supera con creces la ofrecida por el archivo WML en caso de, por ejemplo, disponer de 3 gráficos dentro de una misma página (relación 6000/1000).

Todos estos tiempos y datos se refieren única y exclusivamente a la capa superior de aplicación. No se ha estudiado el efecto que tiene sobre las capas inferiores ya que no varía al estándar WAP actual y por tanto no significa ningún cambio. Dentro de estos tiempos no se contempla tampoco el tiempo que le supone al terminal móvil y al servidor el tener que realizar los cálculos criptográficos necesarios, que aumentan el tiempo real de la transmisión, por la misma razón que antes.

En la simulación realizada, el terminal móvil es sustituido por un Pentium II a unos 300 MHz. y la velocidad en el cálculo criptográfico es prácticamente despreciable. El procesador equivalente actualmente en los aparatos móviles es de un Pentium I a 100 MHz, aunque ya están saliendo nuevas tarjetas para teléfonos móviles más avanzadas y por tanto, la velocidad de procesador será prácticamente la misma que en la simulación.

A continuación, en la figura 3.9 y en las tablas 3.14 y 3.15 se pueden ver unas medidas realizadas de diferentes páginas WML con tamaños entre 2 y 100 Kbytes.

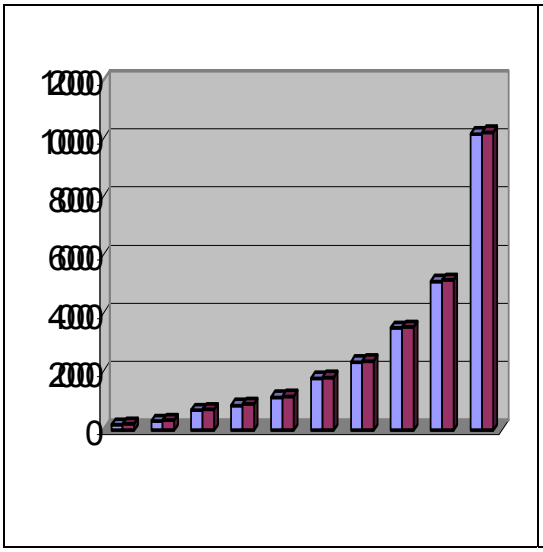


Fig. 3.9 Gráfica de comparación cargas WML claro/WML cifrado.

Página WML inicial (bytes)	Página WML cifrada (bytes)	Aumento de carga (bytes)
1887	2197	310
3117	3421	304
6807	7146	339
8652	8994	342
11387	11722	335
17877	18247	370
23412	23783	371
35097	35492	395
51200	51620	420
101888	102400	512

Tabla 3.14 Datos de comparación de cargas WML claro/WML cifrado.

El aumento de carga que añade la cifrado con TLS es del orden de 300-400 bytes. Cabe recordar que las páginas WML actuales disponibles en los servidores no superan los 10 Kb, y por tanto, en este rango de cargas, el aumento que se tiene es de 300 bytes, o lo que es lo mismo, a una velocidad de GSM de 9600 bps un cuarto de segundo. Sin embargo también se hicieron medidas del tiempo que tardaba en hacer los cálculos criptográficos en las páginas de más de 15 Kb, ya que para las más pequeñas la precisión era bastante mala debido al retardo que introducía la máquina de simulación que era un Pentium II a 300 MHz. En la figura 3.24 se pueden ver estos tiempos:

Página WML cifrada (bytes)	Retardo de transmisión a 9600 bps	Retardo total
18247	15 seg.	29 sag
23783	20 sag	33 sag
35492	29 sag	59 sag
51620	42 sag	1min 23 seg.
102400	1 min. 23 seg.	3 min. 4 seg.

Tabla 3.15 Retardos en la transmisión.

Se puede ver claramente que para el caso de páginas muy extensas el retardo que se acumula es mucho mayor que el propio de transmisión.

Capítulo 4

Usabilidad en WAP.

4.1 Introducción.

En las transacciones de comercio electrónico sobre entornos de móviles, el usuario en movimiento busca contactar con un proveedor de servicio que generalmente se encuentra en Internet. Este proceso resulta lento y difícil por problemas relacionados con el tiempo de respuesta y la poca facilidad de uso desde el dispositivo móvil.

En [NIEL00], [DURL00], [KEAR 01], se detallan resultados empíricos respecto a la usabilidad de los dispositivos WAP. Aunque son optimistas en cuanto al éxito del despliegue de las tecnologías de móviles, los estudios sobre la conveniencia de entregar contenidos con WAP ponen de manifiesto que los usuarios no están satisfechos con las bajas velocidades y la interfaz WML/Script. De acuerdo con varias consultoras especializadas, otros servicios que gozan de aceptación entre los usuarios son el correo electrónico móvil, y el servicio de mensajería SMS. Por lo tanto, la entrega de información a los usuarios debería realizarse al menos en estas modalidades.

La lentitud de la respuesta se debe principalmente a las limitaciones existentes en el canal inalámbrico y en los dispositivos móviles. En consecuencia, es necesario minimizar el volumen de tráfico de datos que circula por este medio, y a la vez procurar que dicha información sea lo más útil posible al usuario de móvil [VETT 01], [JIAN 99].

De lo expuesto anteriormente, se deduce que resulta indispensable ofrecer al usuario los mecanismos que le faciliten el uso y ahorren tiempo en la búsqueda, acceso a la información y contacto con posibles proveedores de servicios. Estos mecanismos se pueden integrar en un intermediario, ubicado convenientemente del lado de Internet en donde pueda estar permanentemente conectado, con anchos de banda de órdenes de magnitud mayores al canal inalámbrico y con mejores recursos en infraestructura y herramientas. Así mismo, puede aprovecharse el intermediario para ofrecer varios mecanismos complementarios que optimicen el tráfico por el canal inalámbrico y provean seguridad de la información transmitida entre el móvil e Internet.

4.2 Diseño del intermediario.

Los objetivos básicos del intermediario apuntan a facilitar la búsqueda y el intercambio de información como también a reducir el tiempo de respuesta. Para cumplir con estos objetivos, el intermediario utiliza mecanismos de búsqueda, almacenamiento intermedio y distribución de contenidos. La figura 4.1 muestra la ubicación del intermediario en este entorno.

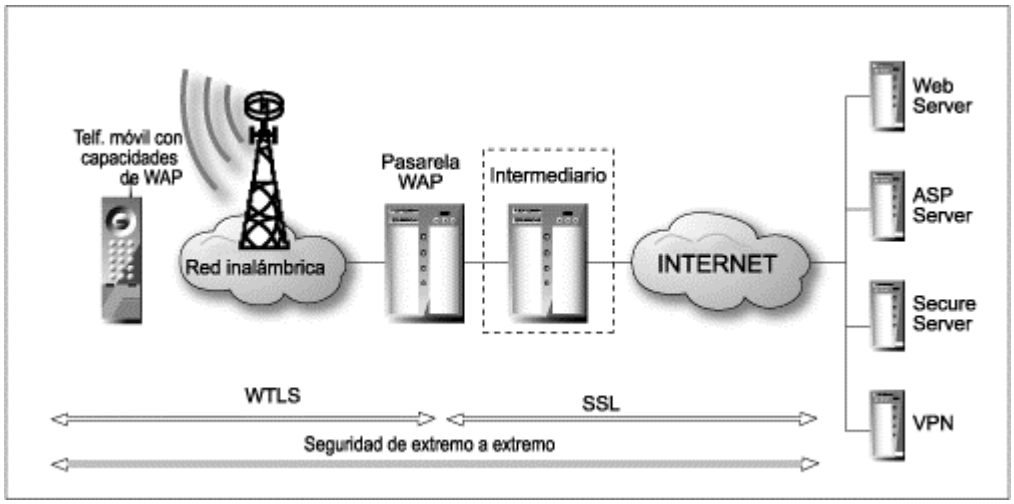


Fig. 4.1 Ubicación del Intermediario.

El entorno de móviles está limitado por las restricciones que le imponen el canal inalámbrico y las características de los dispositivos móviles. La teoría de las restricciones (Theory of Constraints) [GOLD 92], sostiene que cada sistema está sujeto a por lo menos una restricción, que evita que el sistema logre niveles infinitamente altos de desempeño. La alternativa es mejorar la gestión de estos "cuellos de botella", aprovechándolos al máximo. Para maximizar el rendimiento del entorno, se plantean las siguientes estrategias:

Reducir el tráfico de datos en el canal inalámbrico: El canal inalámbrico entre el usuario y la pasarela WAP constituye el "cuello de botella" en cuanto al ancho de banda, por lo tanto deberá en lo posible aprovechar su utilización sola para el tráfico útil. Este objetivo se puede conseguir delegando tareas tales como la búsqueda, interacción y filtrado de la información al Intermediario aún estando fuera de línea.

Reutilizar contenidos: reutilizar la información residente en el almacén intermedio mediante un sistema distribuido de caches y proxies. Esta estrategia permite reducir el tiempo de espera.

Personalizar la información: conociendo previamente el perfil del usuario se puede filtrar y distribuir información en forma predictiva.

La búsqueda de información se delega al subsistema de agentes inteligentes, la información obtenida se organiza en subsistemas de bases de datos para su reutilización mediante caches y proxies, la distribución de información se personaliza utilizando herramientas de gestión de la relación con el cliente, CRM (Customer Relationship Management). Para la búsqueda de información se utiliza un sistema multiagente [MAES 91], [BARB 00], [BLAK 00], [BRAT 98], [CANN 00].

A grandes rasgos, podríamos clasificar la entrega de información en entornos móviles en dos modalidades. Por un lado, cuando un usuario solicita una determinada información, se procesa la solicitud comenzando con la búsqueda de la información que luego será organizada,

filtrada y entregada al terminal WAP, tomando las medidas de seguridad necesarias. La otra modalidad (modo Pull) consiste en gestionar y entregar automáticamente información en forma predictiva utilizando la información del perfil del usuario.

La figura 4.2 presenta el paradigma de las modalidades de Pull y Push del WAP. Utilizar los mecanismos Push implica menor utilización del canal inalámbrico que los pedidos del usuario, que se gestionan en modalidad Pull (viaje de ida y vuelta).

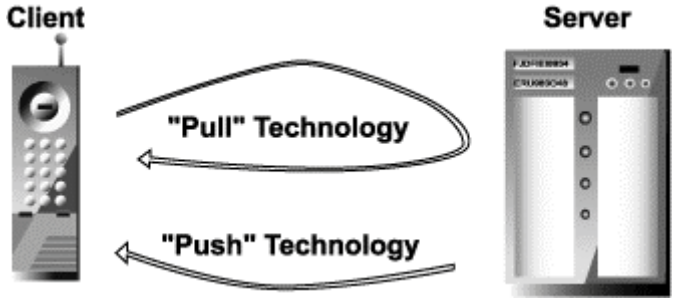


Fig. 4.2 El paradigma de Push y Pull.

La entrega de contenidos en modalidad Push, requiere la generación de una lista de distribución a partir del perfil del cliente. Los contenidos recientes del almacén intermedio se entregan a cada lista mediante un procesamiento por lotes. Un cliente puede pertenecer a varias listas.

En cuanto al almacenamiento intermedio, y más aún la gestión de la información que contiene existen esquemas que han tenido éxito en el manejo y replicación de contenidos en sistemas distribuidos Web mediante la organización de la información en sistemas de caches y proxies [KIST 98], [LOON 97].

Personalizar y por lo tanto, filtrar y reducir el volumen de información que llega al usuario en forma predictiva, implica realizar la gestión de la relación del cliente mediante herramientas CRM, minería de datos y sistemas expertos [ECRM 00], [DURL 99]. Para evitar inundar al cliente con información, se filtran los contenidos del almacén intermedio mediante reglas, en este tema existen los mecanismos adecuados de razonamiento inductivo difuso. Una vez filtrada la información, se la prepara en el formato de entrega adecuado para el cliente (WML/Script, SMS, e-Mail, FTP,...).

La gestión de la relación con el cliente desde el punto de vista de integración del sistema CRM en el Intermediario requiere una estrategia para el arranque, el crecimiento y establecimiento del sistema. Se debe garantizar el cumplimiento de los siguientes parámetros:

- Disponer de las capacidades de inteligencia y análisis para capturar y convertir los datos del cliente en información útil.
- Poder procesar la información proveniente de diferentes fuentes de interacción del cliente por ejemplo: e.mail, Web, SMS….
- Manejar las transacciones desde el Web de la compañía.
- Proveer de un repositorio centralizado de la información de los clientes.
- Integrar en su flujo de trabajo toda la información de manera que esté disponible en cada paso del proceso.
- Funcionar con su sistema operativo y con otras aplicaciones.

Una vez que el intermediario ha realizado el contacto entre el cliente y el servidor es posible que se requiera un canal seguro para la negociación y el pago, en cuyo caso el intermediario

deja de intervenir, y se utiliza la transmisión de datos en modalidad segura extremo a extremo. Se debe considerar la implantación de sistemas de marcas de tiempo o relojes vectoriales así como la utilización de autoridades de certificación para garantizar la confianza de las partes y del mensaje [RAYN 96]. La figura 4.3 presenta el esquema general de los componentes del intermediario.

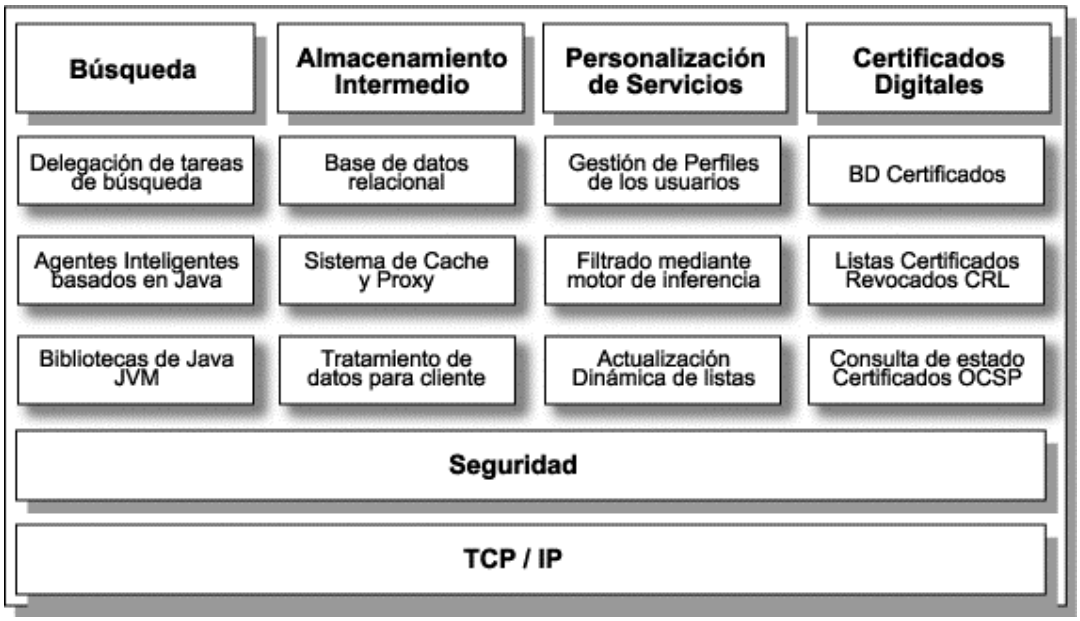


Fig. 4.3 Modelo general del Intermediario.

La figura 4.4 presenta el diagrama de flujo de una solicitud realizada por el cliente WAP. En segundo lugar, la gestión y entrega automática de contenidos utilizando la información personalizada del perfil del usuario. La figura 4.5 presenta el flujograma de la distribución de contenidos para clientes WAP en modalidad Push.

El e-marketing permite segmentar el mercado y personalizar los servicios con la información suministrada por los mismos usuarios. Actualmente está muy ligada a las ventas y se pone especial atención en la fidelización y el servicio personalizado al cliente, basándose en un perfil del usuario, su actualización dinámica y el análisis de sus necesidades. La integración de tecnología CRM permite extender las capacidades del equipo móvil, y facilita su utilización, la gestión se realiza de forma predictiva y dinámica en lugar de bajo demanda y explota las facilidades Push del protocolo WAP, entrega contenidos de posible utilidad para el destinatario. Utilizar los mecanismos Push implica menor utilización del canal inalámbrico que los pedidos del usuario, que se gestionan en modalidad Pull (viaje de ida y vuelta).

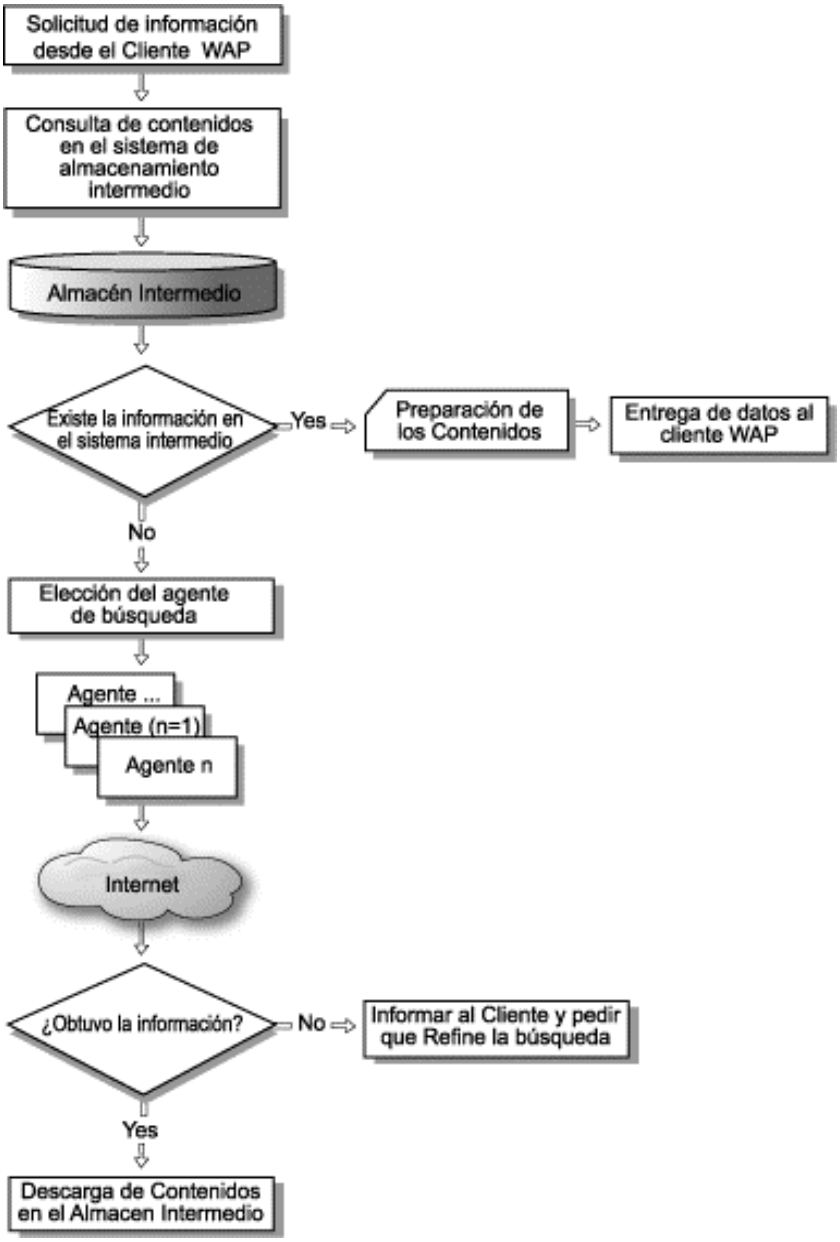


Fig. 4.4 Flujograma de una solicitud (Pull) en entorno WAP.

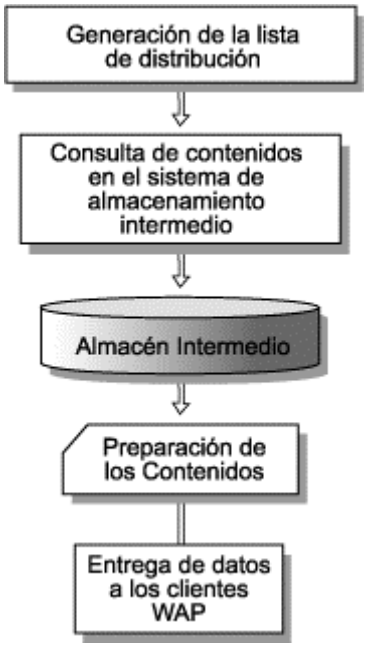


Fig. 4.5 Entrega de información en modalidad Push utilizando el perfil de los usuarios.

Un sistema intermediario multiusuario seguro que gestione la transacción entre el cliente y los proveedores debe integrar sistemas inteligentes que realicen tareas tales como: buscar, aconsejar, contactar, comparar, filtrar, facilitar… . La información de los servidores se almacena y organiza en nuevos mecanismos de bases de datos más complejos que generan contenidos en forma dinámica.

Los agentes inteligentes en el comercio electrónico ya se utilizan para buscar contenidos en Internet y facilitar al usuario la interacción desde su equipo móvil. Sin embargo, esta tecnología requiere del entorno lo siguiente: accesibilidad, determinismo entornos predecibles estáticos / dinámicos, y reglas de acceso. Desafortunadamente Internet no es, un entorno suficientemente amigable para los agentes inteligentes.

Una vez que el intermediario ha realizado el contacto entre el cliente y el servidor, es posible que se requiera un canal seguro para la negociación y el pago, en cuyo caso el intermediario deja de intervenir, y se utiliza la transmisión de datos en modalidad segura extremo a extremo. Se debe considerar la implantación de sistemas de marcas de tiempo o relojes vectoriales así como la utilización de autoridades de certificación para garantizar la confianza de las partes y del mensaje.

Existen propuestas para extender el estándar TLS. En [BLAK 00] "Wireless Extensions for TLS" se propone entre otras cosas, la posibilidad de referenciar un sitio mediante su URL donde encontrar los certificados del cliente, de forma que se descarga al equipo móvil de mantener los certificados del cliente. La autoridad de certificación del intermediario, deberá mantener los certificados de cada cliente, actualizar la lista de certificados revocados (CRL) y gestionar la información del estado de los certificados con OCSP (On-line Certificate Status Protocol).

4.3 Sistema de Búsqueda.

Internet ofrece grandes volúmenes de información extraída de fuentes diferentes. La variedad hace que el acceso y manejo de esta información sea complicado tanto para usuarios

móviles como para usuarios que acceden directamente desde Internet, resultando imposible extraer el máximo de ella. Esta característica de la distribución y disposición de la información hace conveniente la utilización de algún intermediario que facilite el acceso a la información deseada, esto es, sistemas de indexación y motores de búsqueda de información.

El sistema de acceso a la información actual se puede considerar dividido en dos capas: Por un lado nos encontramos con los proveedores de información y por otro lado con los usuarios. Esto implica ciertas dificultades relacionadas con la dispersión y variedad de la información. Para ayudarnos en la tarea de búsqueda de una información concreta, existen buscadores y meta-buscadores que realizan la búsqueda por nosotros. Sin embargo, estos buscadores implican interacción con el usuario, debiendo conocer cómo interactuar con el buscador, definir y formular concretamente la búsqueda expresando la petición correctamente y navegando por la Web.

Entre ambas capas, se ubica el intermediario, en el modelo de tres capas, se organiza la capa intermedia por tipo de actividad, se identifica y vincula el proveedor de información con esta actividad, y el usuario final es vinculado al buscador por actividad. La búsqueda se realiza a partir de un sistema multiagente. El esquema de la figura 4.6 representa una visión de este modelo.

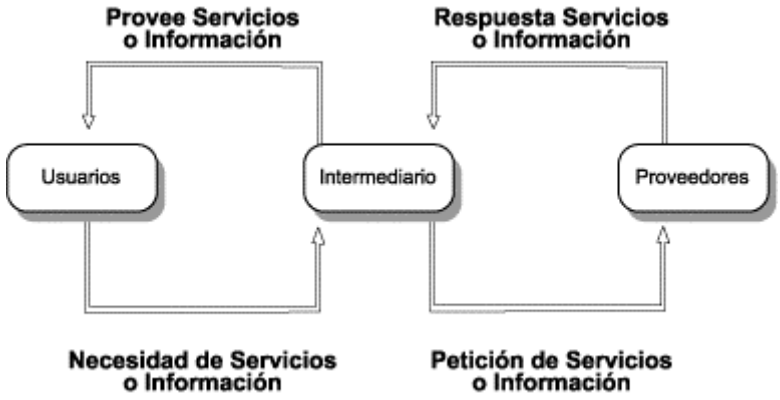


Figura 4.6. Modelo de 3-capas. (ojo)

Se ha de definir un marco de trabajo orientado a proveer información en forma de servicios al usuario móvil final. Si analizamos el diagrama anterior, podemos identificar las tres capas o bloques. El usuario y los proveedores se ven ahora conectados por un intermediario, encargado de registrar las necesidades del usuario final y satisfacerlas mediante la comunicación con los diferentes proveedores. Con la utilización de esta arquitectura, el usuario final no necesita conocer los proveedores y el cómo acceder a éstos, sino que únicamente ha de identificar sus necesidades en el intermediario. El intermediario, basándose en estas necesidades, contactará con los proveedores correspondientes y dará una respuesta a la petición del usuario.

Utilizando esta estructura, resulta más fácil ampliar las fuentes y formas de información, ya que no hace falta que el usuario cambie sus hábitos, sino únicamente, introducir nuevos conocimientos de éste en el intermediario.

4.3.1 Funciones del Sistema de búsqueda del Intermediario

Un sistema multiagente realiza trabajo cooperativo entre sus agentes. La utilización de agentes, nos permite realizar las búsquedas dependientes del contexto donde las realicemos. La característica de los agentes, en cuanto al aprendizaje, hace posible que mejoren las búsquedas

y, unida a la capacidad de comunicación con otros agentes, la refinen. El sistema es capaz de trabajar con entornos dinámicos e información en tiempo real. La utilización de un sistema de agentes, nos permite realizar una búsqueda concurrente y cooperativa entre diferentes agentes por un objetivo común, proveer al usuario con la información requerida. Con la utilización del intermediario, los proveedores pueden ofrecer más información que la indexada por los buscadores.

El intermediario deberá realizar una serie de funciones, tales como:

- Modificar dinámicamente la información pedida por el usuario.
- Manejar la variación, tanto en el número como en la localización, de los proveedores.
- Unificar y procesar las respuestas de los proveedores de acuerdo con el pedido de cada usuario.
- Notificar a cada usuario acerca de los cambios en la información suministrada o cuando la información solicitada se encuentra disponible.
- Entregar en forma asíncrona de la información pedida.
- Actuar como informador acerca de posibles proveedores.

4.3.2 Requisitos del sistema de búsqueda.

El sistema de búsquedas del intermediario deberá cumplir los siguientes requisitos:

- Garantizar la conectividad con los otros módulos del intermediario global.
- Dar respuesta a una serie de peticiones del usuario identificadas como servicios.
- Utilizar un sistema de cache y proxy para la reutilización de la información.
- Permitir la concurrencia en las búsquedas para mejorar los tiempos de respuesta.
- Filtrar de la información obtenida.
- Implementar diferentes formas de entregar la información al usuario.

4.3.3 Diseño del sistema multiagente de búsqueda

El sistema de búsqueda utiliza los siguientes tipos de agentes:

- ***Agentes Inteligentes:*** Los requisitos que deben cumplir los agentes inteligentes incluyen la percepción, el razonamiento y la habilidad de tomar decisiones. Los sistemas de multiagente tienen además sus propios requisitos para comunicar, cooperar y competir en este entorno [MAES 98],[WOOL 96], [WOOL 97].

Las filosofías de diseño de agentes basados en creencia, deseo e intención [BRAZ 97], están entre los más adecuados al objetivo de mejorar comunicación inalámbrica. El aprendizaje por refuerzo se puede usar para aprender las secuencias de operaciones requeridas para lograr un objetivo. El aprendizaje se usa para mejorar el desempeño de agentes. Los ambientes típicos de Internet contendrán una gran cantidad de incertidumbre en el dominio del problema, por lo tanto se puede utilizar las redes Bayesianas, métodos con factor de certeza y métodos de planificación para ejecutar la serie compleja de acciones[MITC 97]. KQML (Knowledge Query Mark-up Language) y FIPA ACL [BLAK 00b], [FIPA 01] proporcionan una estructura para cambiar información y conocimiento entre agentes.

- Agentes Móviles: Los agentes móviles son entidades de programa que pueden transitar libremente la red. Los agentes móviles pueden contener hilos de proceso y, por lo tanto, estar activos y pueden mantener información de su estado y tomar decisiones inteligentes. Difieren de los applets en que ellos pueden transitar independientemente entre diferentes entornos y no están limitados a ser cargados una vez desde el servidor al cliente. Un aspecto importante es la protección del código y de los datos del agente.

Los datos se pueden organizar en una estructura de árbol de decisión, una vez cargados en el equipo móvil, de forma que es posible la interacción fuera de línea de usuario. La interacción fuera de línea es muy deseable en un enlace de comunicación débil, con ruido y con desconexión imprevisible.

- Agentes estáticos: La elección de movilidad o no del agente viene relacionada con su funcionalidad. Un agente que reciba las peticiones del usuario y que no tenga tareas adicionales, no necesitará de las propiedades de movilidad. Éstos son configurados como agentes estáticos.

Los Aglets [BIGU 97], [VENN 97] son ejemplos de herramientas de lenguajes que se pueden usar para desarrollar agentes móviles. Hay una gran variedad de sistemas y arquitecturas de gestión, escritos en varios lenguajes de programación y ejecutables en distintas plataformas, compatibles o no con las especificaciones incluidas en las MASIF y con diferentes aspectos de seguridad. El OMG como autoridad de nombres relativa a agentes móviles incluye explícitamente en las MASIF los siguientes identificadores:

- Lenguajes de programación: Java, Tcl, Scheme y PERL.
- Sistemas comerciales: Aglets, MOA y AgentTcl.
- Métodos de serialización: Serialización de Objetos Java (JOS).

En este punto, realizamos un análisis de la funcionalidad de los distintos agentes, necesarios para el correcto funcionamiento del intermediario. Analizamos tanto su movilidad, su funcionalidad y características principales.

4.3.3.1 Componentes funcionales del sistema de búsqueda.

-*Agente Receptor Peticiones:* Agente estático que se encargará de recibir las peticiones del usuario. Estas peticiones serán comunicadas al agente clasificador. Una vez que el agente receptor de peticiones haya comunicado una petición deberá mantenerse a la espera de nuevas peticiones.

-*Agente Clasificador Peticiones:* Este agente recibirá la petición del usuario y, basándose en su conocimiento, deberá clasificar el tipo de petición. Una vez realizada la clasificación, deberá entregar la información al Agente Dispatcher o programador de Agentes para que programe los agentes necesarios para realizar la tarea requerida.

-*Agente Acceso Base de Datos:* La información local del intermediario, es decir, el conocimiento de éste, estará localizada en una base de datos local. El acceso a ésta estará limitado a este agente especializado, por lo que cualquier agente que necesite de alguna información local, deberá comunicarse con el Agente Acceso Base de Datos. Podemos distinguir entre dos tipos de información: la información de conocimiento y la información de los agentes. Para acceder a estos dos tipos de información, el agente delega la tarea en dos agentes: Agente Acceso Base de Datos Conocimiento y Agente Acceso Base de Datos de Agentes.

-*Agente Acceso Base de Datos Agentes:* Este agente se encargará de acceder a la base de datos de agentes. Podríamos limitar el acceso a éste y que quien quisiera acceder a esta base de datos lo tuviera que hacer a través de él.

-Agente Acceso Base de Datos Conocimiento: Este agente se encargará de acceder a la base de datos de conocimiento. Podríamos limitar el acceso a éste y que quien quisiera acceder a esta base de datos lo tuviera que hacer a través de él. En esta base de datos, podríamos incluir la información de antiguas peticiones de búsqueda de información o preferencias de los usuarios.

-Agente Lanzadera de Agentes: Este agente se encargará de programar los diferentes agentes según la clasificación de la petición realizada por el Agente Clasificador Peticiones. Una vez programados, les dará la información necesaria y lanzará la ejecución.

-Agente Receptor Resultados: Este agente, de tipo estático, recibe los resultados de las búsquedas entregados por los distintos agentes móviles.

-Agente Filtrador Resultados: En caso de que sea necesario, este agente tendrá la tarea de filtrar los resultados obtenidos y presentarlos. Esta característica de filtrar los resultados, podría incorporarse en cada uno de los agentes móviles. Entendemos como filtrar a la acción de extraer la información importante de los resultados obtenidos, obviando el resto.

-Agente Presentación Resultados: A partir de la información filtrada entregada por el Agente Filtrador Resultados, este agente deberá presentar la información obtenida para poder ser entregada al usuario. La información podrá ser entregada al usuario en forma de página WML o en forma de mensaje SMS, con lo que habrá que limitar su contenido y extensión.

-Agentes Específicos Buscadores: Dispondremos de una serie de agentes que serán específicos para un buscador determinado, dotados del conocimiento y la habilidad para acceder a los buscadores, lanzar peticiones y recoger los resultados. A partir de los resultados obtenidos de los distintos buscadores deberemos unificarlos y presentarlos conjuntamente, evitando resultados redundantes. El funcionamiento del sistema se esquematiza en la figura 4.7.

4.3.4 Seguridad del Sistema multiagente de búsqueda.

El sistema de búsqueda basado en agentes requiere protección tanto del código como de los datos. La aplicación residente en el intermediario estará bajo la seguridad de un cortafuegos, es decir, colocaremos un cortafuegos entre nuestra aplicación e Internet, diferenciando entre dos zonas: La zona interna o segura y la DMZ o zona desmilitarizada.

Podemos distinguir entre los agentes estáticos y los agentes móviles. Estos últimos accederán a Internet, pasando por el cortafuegos y utilizando el protocolo ATP. Se utiliza el Protocolo de Transferencia de Agentes para enviar el agente al servidor remoto. Hemos de controlar el acceso por parte de otros agentes a nuestro servidor de agentes. Por política de seguridad, solo permitiremos que entren los agentes móviles que hayan sido generados por nuestro intermediario.

Teniendo en cuenta que los agentes móviles no llevarán consigo información confidencial del usuario, hemos de garantizar la integridad del agente. Esto es, garantizar que la información que lleva el agente no ha sido modificada durante su viaje. Para esta labor podemos utilizar funciones criptográficas, utilizando resúmenes o funciones de hash.

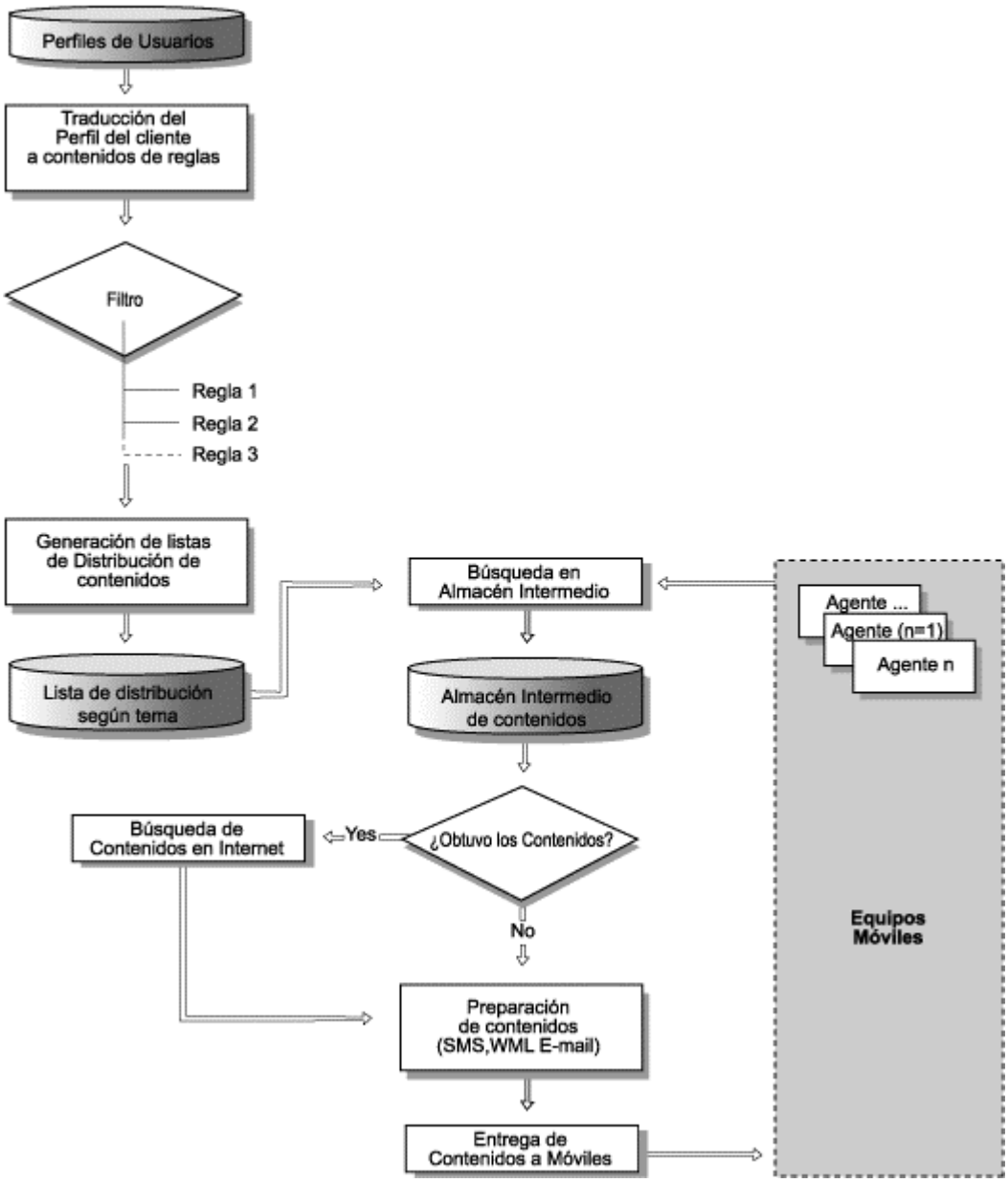


Figura 4.7 Esquema de Funcionamiento del sistema.

4.3.4.1 Seguridad de los agentes móviles.

Un agente móvil es un programa que viaja de un ordenador a otro a igual que los virus. Los sistemas deben hacer hincapié en la seguridad de los ordenadores y del propio sistema multiagente. Los aspectos de seguridad típicos que deben ser controlados son:

- Protección de la máquina contra los agentes.
- Protección de los agentes contra la máquina.
- Protección de la red.

Los ataques más comunes que pueden realizarse a un sistema de agentes móviles son:

- Inundar el sistema con peticiones, tanto legales como ilegales.
- Escuchar la red para obtener información privada.
- Modificar, borrar o sustituir cualquier elemento transferido por la red.
- Grabar y retransmitir ilegalmente una comunicación.
- Falsificar la identidad – enmascaramiento - de un agente o sistema de agentes para tener acceso a la información o a ciertos servicios.
- Utilización abusiva de algún recurso para que no pueda ser utilizado por otro usuario.
- Colocar un Caballo de Troya - agente o sistema de agentes - para recibir información confidencial o denegar acceso a los recursos.

Tanto los sistemas como los propios agentes móviles deben reforzar las tareas de seguridad para evitar, de un modo fiable, los ataques descritos anteriormente. Pueden tener varias políticas de seguridad que permitan:

- Comprobar las credenciales de los participantes en cualquier operación.
- Restringir o garantizar las operaciones que puede ejecutar un agente.
- Gestionar privilegios de acceso a los recursos y establecer límites de consumo.

Los requisitos que deben garantizarse en cualquier comunicación son:

- Confidencialidad: evitar la escucha del canal.
- Integridad: comprobar que los datos no han sido modificados durante la transferencia.
- Autenticación: tanto el agente o sistema emisor como el receptor deben ser identificados para evitar accesos a información o a recursos reservados.
- Detección de reproducción: evitar la duplicación de un agente durante una comunicación.

Aunque hay una gran variedad de políticas de seguridad que pueden utilizarse para evitar los ataques, un proceso de comprobación típico antes de iniciarse el viaje de un agente incluye los siguientes aspectos:

- El sistema debe verificar la autoridad propietaria del agente.
- Durante la creación de la petición, el propietario define las preferencias de seguridad para el agente.
- Al crearse la instancia del agente, se incluye información sobre su autoridad y la de su sistema.
- El sistema origen codifica la información.
- Los sistemas origen y destino crean un canal de comunicaciones seguro.
- El sistema destino descodifica la información y realiza las comprobaciones necesarias.

4.4 Sistema de almacenamiento intermedio.

El intermediario tiene unas funciones de almacenamiento intermedio con las que se pretende agilizar el acceso de los usuarios a los contenidos de la Web, a través del mantenimiento de un caché de documentos. La información obtenida mediante de los mecanismos de búsqueda inteligentes es entregada a los clientes, pero también es clasificada y recogida en un almacén intermedio de documentos común a todos los usuarios.

El intermediario actuará como un servidor-proxy y atenderá todas las peticiones de los usuarios. Al recibir estas solicitudes, deberá comprobar en cada caso si una versión actualizada del documento se encuentra disponible en él caché y si es así, entregará al cliente esta

información guardada en memoria. En este caso, la respuesta del intermediario es mucho más rápida porque no requiere delegar las tareas de búsqueda a otros sistemas. Sin embargo, si el documento no está disponible, la petición será enviada directamente a los mecanismos de búsqueda con agentes inteligentes.

En caso de desconexión eventual entre el intermediario y el resto de la red TCP/IP, el subsistema de caché es el único responsable de la entrega de documentos. En esta situación, el intermediario trabajará en modo de desconexión y sólo será posible el acceso y la distribución de contenidos previamente almacenados en memoria. Así mismo, mediante el uso de agentes periódicamente se deberán refrescar las páginas en memoria.

4.5 Sistema de personalización del servicio.

Últimamente algunos autores han estado trabajando en temas sobre asociación de Internet y CRM, y desarrollando conceptos como IIM (Internet Interaction Management), ICRM (Internet Customer Relationship Management), o e-CRM (Electronic Customer Relationship Management). Todos estos procesos tienen un significado similar y persiguen el mismo fin. El objetivo principal de la aplicación de Internet al CRM es lograr, a través de una plataforma basada en Internet, la lealtad y satisfacción de los clientes haciendo uso de transacciones on-line.

La ventaja principal de Internet en este entorno es que, siendo considerada una herramienta
de contacto entre la empresa y los clientes, no sólo comunica sino que permite interactuar y transferir información en ambos sentidos. Las interacciones entre empresas y clientes conducidas a través de Internet, aplican idealmente el proceso del CRM, porque permiten la utilización de una plataforma tecnológica para el diseño, desarrollo e implementación de técnicas de gestión de información, como el data warehousing y el data mining.

También los agentes inteligentes han empezado a jugar un papel muy importante en esta evolución, ya que pueden usar técnicas para conocer los canales y productos óptimos que se necesitan combinar para mejorar la experiencia del consumidor. Entre otras cosas, estas técnicas se utilizarán para valorar el impacto de la lealtad del consumidor en factores como el coste o la calidad del servicio y aplicando correctamente los algoritmos a las oportunidades de mercado, las empresas pueden crear ventajas competitivas en Internet.

Por otro lado, uno de los requerimientos básicos en el proceso CRM es disponer de la información de los clientes de una forma rápida y fácilmente accesible para la empresa. Los componentes necesarios para implementar estas soluciones incluyen una interfaz con el usuario y una base de datos. La implementación debe ser segura, con total accesibilidad para la empresa y construida como un sistema abierto que pueda ser usado por múltiples aplicaciones de negocios. La utilización de Internet para el CRM supone una considerable transferencia de datos, por tanto requiere mecanismos que permitan al consumidor interaccionar con la empresa de forma segura.

En este caso, la utilización de Internet para la obtención y el análisis de datos de los clientes permite hacer eficiente y eficaz el proceso de CRM. Internet facilita estas tareas mediante una página Web, que puede estar conectada automáticamente a una base de datos donde se almacena la información obtenida. Además, se logra otro de los objetivos fundamentales del CRM, como es la personalización de los servicios o productos. Las tecnologías asociadas a Internet tienen la facilidad de asociarse con sistemas automatizados de telemarketing, donde se procesan consultas, observaciones y quejas de los clientes, que se almacena en una base de datos para su posterior análisis. Internet
no sólo aporta mecanismos para la atención al cliente, sino que es capaz de ayudar en los procesos de compra, abastecimiento y producción, y por tanto puede ofrecer una solución integral a las empresas.

Así pues Internet facilita el desarrollo de técnicas CRM porque permite la obtención de información de los clientes de una forma ágil y puede mejorar la relación con los consumidores haciendo uso de herramientas e-business y de aplicaciones como el e-mail. Internet es el medio ideal para la implementación de una estrategia de CRM, pero es recomendable complementarse con otros medios para no perder otros canales de contacto con los clientes.

La personalización de servicios se consigue mediante la gestión de la relación con el cliente basándose en el perfil del usuario, en su actualización dinámica y en el análisis de sus necesidades. El perfil del usuario no es más que el registro de información sobre un cliente en una base de datos, que incluye, principalmente, los hábitos y preferencias del usuario. Este perfil clasifica al cliente vinculándolo a ciertas listas de distribución y se actualiza dinámicamente cada vez que el usuario realiza una petición de servicio. Esto significa mejoras en la usabilidad, porque permite extender las posibilidades del equipo móvil y facilita su utilización.

En la modalidad de servicio Push, el perfil es utilizado para determinar a que lista de distribución pertenece cada cliente. Los contenidos serán entregados adecuadamente a los componentes de cada lista de distribución con una frecuencia concretada por cada uno de ellos. En cambio, cuando se realiza la búsqueda de información bajo pedido, el perfil del usuario ayudará a determinar de entre una amplia variedad de resultados, los más adecuados para ser entregados al cliente según sus necesidades.

4.6 Certificados (OCSP).

Para la distribución de las claves públicas en entorno abierto, la solución que más ampliamente se ha adoptado, consiste en recurrir a una tercera parte confiable, llamada autoridad de certificación (CA), propuesta por primera vez en 1978 por Kohnfelder. Las funciones de una CA consisten en verificar la identidad de solicitantes de certificados, crear los certificados y proporcionar los mecanismos necesarios para comprobar la validez de los certificados emitidos. El tema de la revocación ha sido objeto de diferentes estudios [COOP 01], [MICA 96], [MYER 01].

El grupo de trabajo IETF ha desarrollado una propuesta para emitir el estado de certificados llamada On Line Certificate Status Protocol (OCSP). Este protocolo nació en base a dos borradores propuestos [BRANC 98], [MYER 99], actualmente se encuentra en discusión una nueva propuesta [MYER 01]. Mediante este protocolo el usuario recibe el estado del certificado o grupos de certificados que necesita.

Al disponer el intermediario de un repositorio de certificados, se puede consultar su estado de forma ágil, otra ventaja es que el sistema de móviles puede funcionar en forma autónoma si en algún momento no se dispone de acceso a la infraestructura de clave pública. Por otra parte, el protocolo OCSP, es un protocolo ligero adecuado a los entornos de móviles dadas sus características.

4.7 Conclusiones.

Mejorar los entornos con limitaciones implica levantar las restricciones del sistema, o de no ser posible eliminar estas limitaciones, se debe optimizar los mecanismos involucrados.

Una restricción importante que se puede levantar es el ancho de banda, es por esto que se ofrece en las generaciones 2.5G y 3G mayor disponibilidad en el ancho de banda y mejoras en la interfaz de usuario, que como hemos visto son las más importantes.

Los entornos de móviles presentan paradigmas propios, la información geo-referenciada, la oportunidad en la entrega de la información, mientras los usuarios están en movimiento, imponen nuevos retos y paradigmas en el tratamiento de la información.

La desconexión impredecible, hace necesario proveer agentes capaces de interactuar con el usuario fuera de línea. Esto significa una nueva manera de llegar al usuario, es este momento, Java 2 Micro Edition ya ofrece esta posibilidad, mientras que WAP no soporta Java.

Otro factor que pesa al momento de conectarse es el coste, especialmente en GSM. GPRS y los servicios basados en conmutación de paquetes ofrecen la ventaja al usuario de estar permanentemente conectado.

Las operadoras telefónicas disponen de la información del usuario que les permite proveer este tipo de servicios con ventaja sobre la competencia. Sin embargo, para abrir el entorno inalámbrico a Internet es necesaria la compatibilidad con los estándares y servidores ya existentes en Internet.

Es de esperar que la siguiente generación de móviles contemple estos aspectos, aproveche las ventajas del entorno y dimensione sus limitaciones frente al entorno de ordenadores conectados por cable.

Capítulo 5

Evaluación de las prestaciones del sistema global.

5.1 Introducción.

Los entornos inalámbricos de telefonía móvil se caracterizan por ser entornos limitados tanto en recursos de la red como en el propio dispositivo móvil. Una de las más fuertes limitaciones es el canal inalámbrico. Otro problema se refiere al tiempo de espera de una petición. Con el objetivo de gestionar en forma más eficiente estos recursos, en el capítulo anterior se propuso la utilización de un intermediario dotado de mecanismos de búsqueda, almacenamiento intermedio, y personalización de servicios.

Se puede evaluar la gestión del entorno inalámbrico, midiendo los siguientes parámetros:

- Tiempo de respuesta de un pedido.
- Volumen de tráfico de datos sobre el enlace inalámbrico.
- Facilidad de uso.
- Percepción subjetiva del usuario del tiempo de espera.

Para poder evaluar las prestaciones que tendría el sistema global con la presencia de un intermediario se ha procedido a realizar una simulación del comportamiento global. Para poder realizar la simulación del tráfico en el entorno, tal como ilustra la figura 5.1, se ha establecido una similitud con el entorno inalámbrico sin intermediario y se ha ajustado sus parámetros con los obtenidos empíricamente en investigaciones de campo como las presentadas por Nielsen [NIEL 00]. Una vez ajustados los parámetros de operación del modelo, se evalúa el rendimiento con los mecanismos complementarios que provee el intermediario.

5.2 Consideraciones de diseño del intermediario.

Dadas las características de la arquitectura WAP, el tráfico que circulará en estos entornos puede realizarse en dos modalidades: Push, sin pedido explícito del cliente, y aprovechable para la entrega de contenidos en forma predictiva, (por ejemplo: conociendo el perfil del usuario). La segunda modalidad Pull, bajo petición del usuario, implica un viaje de ida y vuelta por el canal inalámbrico, por lo tanto, es deseable que el tráfico que circula sea aprovechado al máximo y a la vez se reduzca el tráfico de datos que pasa por este canal al mínimo posible.

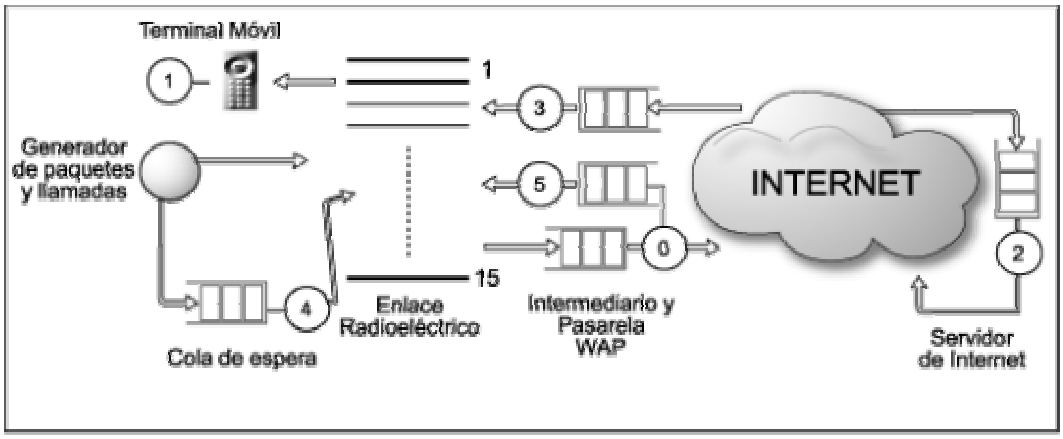


Fig. 5.1 Entorno a evaluar.

5.2.1 Esquema del modelo simulado.

El modelo consta de las siguientes partes:

a) Modelo de fuente:

Se trata de un generador de paquetes de datos y llamadas de voz, que emula a N usuarios multiplexados haciendo consultas a través de WAP o realizando llamadas de duración aleatoria, teniendo en cuenta que los recursos son limitados y que la voz tiene prioridad sobre los datos.

Las llamadas y los paquetes generados son caracterizados por procesos de Poisson, es decir, se asume que las llegadas son poissonianas con unas tasas independientes entre ellas, y que irán en función del tráfico ofrecido al sistema y del tiempo entre llegadas (siendo el tráfico ofrecido y el tiempo entre llegadas diferente para llamadas y para paquetes de datos).

Dado que el modelo del enlace radioeléctrico emula los quince canales que podemos encontrar en GPRS (dos portadoras), como máximo podrá darse servicio a 15 comunicaciones simultáneas. Estos canales podrán contener tanto llamadas de voz como paquetes de información, tanto en sentido de subida como de bajada, es decir, tanto en la dirección usuario - pasarela WAP como en la dirección pasarela WAP - usuario. Todo el tráfico que se va generando se introduce en el enlace radioeléctrico, pero se puede dar el caso que no tengamos recursos suficientes en el momento de las nuevas llegadas.

En el supuesto de que nos llegue un paquete de datos y no haya canales disponibles se introduce en la cola del enlace radioeléctrico, en espera de tener algún canal libre. Si lo que llega es una nueva llamada y no tenemos canales disponibles, se saca un paquete de datos que se esté sirviendo (el cual se pone en cola), sacando primero los paquetes en sentido de subida y en última instancia los que van en sentido de bajada (estos también se volverán a introducir a la cola correspondiente); de esta manera la llamada pasará a tener un canal para servirse. Si los quince canales estuviesen ocupados por tráfico de voz, las llamadas no podrían ser servidas. Para establecer el tráfico de voz entrante, se fija cual es la probabilidad de pérdidas que el sistema puede asumir. A partir de ese valor y del número de circuitos, consultando las tablas de Erlang 1 (o Erlang B) se determina el tráfico de voz máximo que se puede ofrecer al sistema para tener esa probabilidad de pérdida. Debe tenerse en cuenta que el tráfico de datos no altera esta probabilidad ya que se da preferencia total al de voz.

Para el tráfico de datos se ha procedido de forma similar; se fija una probabilidad de demora baja y se asume que no hay tráfico de voz. A partir de las tablas de Erlang 2 (o Erlang C) se calcula una cota superior de tráfico de datos para dicha probabilidad de pérdida. En este caso, se trata de una cota superior, ya que el tráfico de voz va a alterar en gran medida este resultado.

El simulador permite obtener estadísticas de paquetes perdidos y llamadas perdidas, de forma que se puede analizar el comportamiento de la fuente, así como su eficiencia. Si llegan voz y datos simultáneamente, se tratan las dos llegadas en función de los recursos disponibles en aquel instante, dando prioridad a la voz. Los paquetes petición (sentido de subida) tendrán todos el mismo tamaño, lo único que diferenciará un paquete de petición de otro será el campo búsqueda del paquete, que no será nada más que un número generado aleatoriamente, acotado entre dos valores conocidos para tener un control sobre las peticiones que se puedan hacer y poder modelar el intermediario adecuadamente, puesto que así podemos ver si su uso es efectivo.

Los paquetes respuesta (sentido de bajada, es decir, las páginas solicitadas), tienen un tamaño variable siguiendo una distribución de Pareto. Esto hará que ocupen los recursos del enlace en sentido de bajada un tiempo variable. Estos paquetes también estarán sujetos a la política de expulsión del enlace radioeléctrico en caso de llegadas de voz sin canales libres

b) Modelo del enlace Radioeléctrico.

Como se ha mencionado anteriormente, el modelo del enlace radioeléctrico emula los quince canales que podemos encontrar en GPRS. La estructura de canales creada está formada por un vector de quince posiciones, en la que cada posición es de tipo canal. El tipo canal contiene información de tiempo de transmisión (tanto de paquetes como de voz), llamadas tratadas, paquetes transmitidos en ambos sentidos e identificadores de paquetes que están siendo transmitidos.

El enlace radioeléctrico posee una cola propia para los paquetes en sentido de subida. Esta cola se irá ocupando por aquellos paquetes de datos que no pueden ser servidos en un determinado momento ya que el enlace radioeléctrico no dispone de ningún canal disponible para ellos. Asimismo, dado que las llamadas de voz tienen prioridad absoluta por encima de cualquiera paquete, cuando llegue una llamada de voz y no tenga ningún canal disponible, introducirá en la cola un paquete de datos que se esté transmitiendo, si lo hubiere. Si llega un nuevo paquete y se encuentra con algún canal de datos disponible, tendrá prioridad frente a los que se encuentran en cola del enlace radioeléctrico, estos paquetes pueda ser que hace rato se esperen, y al final se acaben perdiendo por time out.

Los paquetes de bajada tendrán prioridad frente a los paquetes de subida, puesto que con la política seguida de expulsión por timeout, se optimizan recursos si se transmite primero la información encontrada en Internet que las peticiones pendientes. Los paquetes de subida expulsados pasen a la cola del enlace radioeléctrico (cola 4 en la figura 5.1), junto con las nuevas llegadas que no han encontrado canal de datos disponible, mientras que los paquetes en sentido de bajada pasan a la cola del servidor correspondiente (cola 3 en la figura 5.1)

Bajo un punto de vista temporal, el enlace radioeléctrico constituye el cuello de botella del sistema, es decir, el factor limitador del esquema y el que provocará que los tiempos obtenidos sean grandes en comparación a los que se podrían obtener si no hubiera la interfaz aire de por medio.

c) Modelo de Colas.

El modelo desarrollado consta de cinco colas:

1. Cola del intermediario, para los paquetes procedentes del modelo de fuente.
2. Cola del intermediario, por los paquetes procedentes de Internet.
3. Cola del servidor de Internet.
4. Cola del terminal WAP.
5. Cola del enlace radioeléctrico en sentido de subida.

Todas estas colas acaban en un servidor, uno por cada una de ellas, encargado de servir los paquetes (sacarlos de la cola). Lo que une un servidor con la cola de uno de los otros servidores son enlaces, de diferentes velocidades y tipologías en función de lo que estemos simulando. La ubicación de estos enlaces es entre el modelo de fuente y la pasarela WAP (subida y bajada) y entre el Intermediario e Internet (también subida y bajada).

La implementación de estas colas garantiza que no se pierde ningún paquete (excepto por time out), y nos permite calcular el tiempo de permanencia en el sistema de cada uno de los paquetes.

El modelo de colas implementado tiene en cuenta los elementos en cola de cada servidor, el tiempo de permanencia en el sistema de cada uno de los paquetes, y los retrasos sufridos por los paquetes a lo largo de los enlaces. Para su simulación se ha considerado la cola y el enlace que la precede como un único elemento, de modo que los paquetes siempre están localizados en alguno de los sistemas virtuales existentes (entendemos por sistema virtual el conjunto de servidor + cola + enlace).

Cada cola será servida por un servidor con un tiempo de servicio determinado, que cuantitativamente será despreciable respecto de los tiempos que se manejan. A la cola del intermediario entrarán absolutamente todos los paquetes que se acaben enviando correctamente a través del enlace radio, es decir, los que no se pierden por la política de timeout.

Los enlaces tendrán las siguientes velocidades:

- Enlace 0: 10000 b/s (simula el enlace radioeléctrico de subida, CS-2)
- Enlace 1: 10000 b/s (simula el enlace radioeléctrico de bajada, CS-2)
- Enlace 2: 64000 b/s (simula un enlace de Internet)
- Enlace 3: 64000 b/s (simula un enlace de Internet)

d) Intermediario o Broker.

El intermediario es una base de datos que tiene almacenadas páginas ya consultadas recientemente, agrupadas en función de unos determinados perfiles de usuario. Su ubicación será a una distancia pequeña de la pasarela WAP, puesto que de esta manera no se añade otro retraso por el hecho de tener que redirigir los paquetes hacia otro sitio (podemos considerar que el intermediario está integrado a la pasarela en el caso ideal).

Su funcionamiento es el siguiente: cuando llega una consulta al intermediario este lo que hace es comparar la URL solicitada con las que ya tiene en su base de datos (también lo denominaremos proxy). Si consigue encontrar la petición del usuario, encamina hacia el terminal móvil la página solicitada, ahorrándonos de esta manera la búsqueda a través de Internet, puesto que la tiene en caché. Si la búsqueda resulta infructuosa, lo que haría el intermediario es dirigir la petición hacia Internet haciendo uso de un agente, y esperar una respuesta, es decir, la página solicitada por el usuario.

Cuando el servidor de Internet encuentra lo que se le ha pedido envía la página encontrada hacia el intermediario, y es en ese instante cuando el intermediario se da cuenta que la búsqueda a través de Internet ha tenido éxito. Se añade la referencia y la página URL a su base de datos, de tal modo que en un futuro aquella búsqueda ya no se tendría que llevar a término. En definitiva, lo que hace el intermediario es actualizarse cada vez que se hace una nueva consulta a Internet. Finalmente, la página se dirige al terminal móvil, como en el caso anterior. Por otra parte, las páginas que ya están almacenadas en la memoria del intermediario se irán refrescando dinámicamente cada cierto tiempo, mediante los agentes móviles.

El intermediario no será nada más que un vector que almacenará URLs, las cuales asociará con un tamaño. Si la petición que se está buscando coincide con alguno de los elementos que se encuentran en el proxy lo que hará este es encaminar la respuesta (tamaño aleatorio de respuesta, siguiendo una distribución de Pareto con una media de valor 31341 bytes). Para el cálculo de la distribución de Pareto se ha tomado un factor de forma igual a 1.5549 [REYE 99]. Es necesario decir que el tiempo de proceso dentro del proxy puede considerarse despreciable, porque en todo momento estamos hablando de tiempos muy superiores en orden de magnitud al que tarda el intermediario al procesar el paquete (el orden de magnitud de los tiempos de transmisión del enlace radioeléctrico es muy superior), pero por claridad no se ha omitido ningún parámetro importante desde el punto de vista lógico de funcionamiento del programa.

Si no se encuentra lo buscado, se encamina la petición hacia Internet, que será el caso de simular la búsqueda utilizando un agente, es decir, una búsqueda normal.

e) Entorno de Internet.

Se encarga de modelar una búsqueda normal en Internet. El paquete enviado por el intermediario navega a través de Internet hasta que algún servidor consigue encontrar la URL solicitada, y entonces es él quien devuelve la página al intermediario, con el retraso temporal que todo esto comporta. Este último caso coincide con el funcionamiento actual de las búsquedas usando WAP.

Para modelar correctamente esta parte y simplificarla al máximo hemos considerado un único servidor de Internet, al que se accede por un enlace de velocidad 64000 b/s, que tiene una cola (ver modelo de colas), y que tiene un tiempo de servicio propio. Para simular una búsqueda sobre cualquiera servidor de Internet, así como simular diferentes estados de carga del enlace, se considera un único servidor de Internet y se introduce un factor de aleatoriedad temporal en el tiempo de servicio, de forma que siempre se tarda un tiempo aleatorio en llegar al servidor de Internet, es decir, podríamos interpretar que se está consultando diferentes servidores. La ocupación de la cola del servidor se podría interpretar como si fueran paquetes de otros usuarios.

Dado que el tamaño de página es aleatorio (caso del enlace 3, cuando el que estamos transmitiendo por el enlace es la página encontrada), sigue una distribución de Pareto, el tiempo que se tarda en transmitir una respuesta también lo es. Este valor, dividido por la velocidad del enlace y pasado a bits, es el que se usa como media de una exponencial, que nos dará un valor aleatorio. Para el caso del enlace 2 el tamaño es fijo, para el que el valor medio temporal a considerar en el exponencial siempre será el mismo.

A este tiempo de transmisión del paquete a través de los enlaces de Internet, es necesario añadir otro tiempo aleatorio debido a los retrasos que sufre el paquete por tener que atravesar diferentes redes y nodos, puesto que cada cosa que atraviesa le introducirá un retraso. Para modelar este tiempo correctamente, lo que hemos hecho es utilizar el tiempo de ping, cogiendo un valor medio de tiempo de ping de Internet [GSP 00] (página que se dedica a sacar estadísticas sobre tiempo de ping entre otros cosas, para estudiar el estado Internet) igual a

1.259ms, y utilizarlo como media de la función exponencial. Por lo tanto, con la suma de estos dos tiempos habremos modelado el retraso sufrido por el paquete a través de Internet.

Una vez se ha realizado la búsqueda y ha tenido éxito, lo que se hace es encaminar la página encontrada a través del otro enlace (enlace 3) hasta el intermediario, tal que este tenga constancia que la búsqueda se ha realizado satisfactoriamente y pueda, de esta manera, agregar la URL y la página a su base de datos, para no tener que realizar la búsqueda de nuevo. Para este enlace también se ha considerado un tiempo aleatorio, como hemos comentado anteriormente.

f) Terminal móvil.

Es el destino dónde finalmente llegarán todas las páginas solicitadas. Será el sitio dónde evaluaremos los diferentes retrasos sufridos por cada uno de los paquetes, diferenciando los que han estado servidos por el intermediario directamente o por los que, por el contrario, han estado atendidos por un servidor de Internet.

Al terminal WAP acabarán llegando todos los paquetes, pero se considerará los tiempos particulares de los paquetes encaminados directamente desde el proxy (con identificación 1) y los paquetes que han ido a través de Internet (los que tienen identificación 2), para hacer un estudio de los dos casos y ver el ahorro temporal que supone tener un proxy.

5.3 Funcionamiento del modelo.

El programa implementado simula el retraso temporal que sufren los usuarios a la hora de hacer una consulta a través de WAP, así como la carga que soportan los canales radio del enlace radioeléctrico. Para precisar, este retraso temporal contempla el lapso de tiempo que transcurre desde que sale la petición (paquete petición) hasta que llega la página solicitada (paquete respuesta).

En modalidad Pull, el tiempo de respuesta de una petición implica el camino de subida y búsqueda en Internet y luego el camino de bajada con la información en caso de ser encontrada. Este tiempo es variable, y depende de los elementos de la red Internet y que la información este accesible. Estudios demuestran además que la autosemejanza del canal de bajada es aún mayor que del canal de subida [BELL 00].

Se considera que el tiempo mínimo de una petición, se reduce a encontrar la información en el sistema de almacenamiento intermedio (cache y proxy) del intermediario. En caso de no disponer de la información buscada, el intermediario utiliza el sistema multiagente de búsqueda por Internet. Si la información no se encuentra en Internet, se espera un tiempo de expiración del pedido (time out) transcurrido el que se devuelve el mensaje que indica que la información solicitada no se ha encontrado y se proveen las opciones para redefinir la búsqueda.

Otro caso diferente, lo constituye la distribución de información en modalidad Push, se parte de la información del usuario y de información a distribuir que ya existe en el intermediario. Los problemas planteados en esta modalidad tienen relación con el filtrado, formateado, y generación de listas de distribución, y no son abordados en esta evaluación porque el tiempo de respuesta no es crítico.

Para simular el tráfico, se considera el tráfico de voz con prioridad sobre el tráfico de datos. El tráfico de datos se caracteriza por las ráfagas y por su autosemejanza, se utilizan modelos de fuentes de Gamma-Pareto, Pareto y similares. Sin embargo, el modelo de fuente clásico se basa

en la f.d.p. de Poisson, la que reúne una serie de características que la hacen más tratable numéricamente.

Para fines prácticos, se puede abordar el tema con las funciones clásicas, aunque existen trabajos como los de [BARC 00] [BARC 00b] [BARC 01][CHENG 00][LELA 94] que profundizan este tema.

5.3.1 Diagrama de Bloques.

Para visualizar el funcionamiento del programa se adjunta el diagrama de bloques de la figura 5.2 que explica gráficamente el funcionamiento del mismo.

5.4 Escenarios de prueba. Resultados.

Una vez desarrollado el simulador se trabajó con distintos escenarios y distintas cargas tanto para voz como para datos. Inicialmente se planteó un escenario sin tráfico de datos, es decir, los únicos usuarios del sistema efectuaban llamadas de voz. Obviamente, la presencia del intermediario en este escenario no provoca ninguna modificación en los resultados. Simplemente, se utilizó para, comparando con las tablas de Erlang, poder validar el comportamiento del simulador en cuanto a tráfico de voz se refiere.

Superada esta fase, para poder verificar su correcto funcionamiento al introducir tráfico de datos, se simuló el sistema global con tráfico de voz y datos sin introducir el intermediario y se compararon los resultados obtenidos con el trabajo de campo de Nielsen [NIEL 00], que nos da una referencia empírica de los tiempos que se obtienen actualmente con WAP. Los resultados obtenidos fueron prácticamente idénticos, de forma que se procedió a la introducción del intermediario en el modelo.

A grandes rasgos, las conclusiones que se pueden extraer del análisis de resultados son las siguientes:

- Si el sistema no está saturado, se obtienen mejoras considerables en cuanto al número de paquetes transmitidos y retardo medio extremo a extremo, no-solo de forma global para los paquetes de datos, sino también atendiendo exclusivamente a las solicitudes que no están almacenadas en el sistema de caché y deben buscarse en Internet. Esto es así, porque la cola donde se almacenan los mensajes procedentes de Internet (cuello de botella del sistema) está mucho menos congestionada al haberse gestionado muchos paquetes directamente.

- Si el sistema está saturado, el intermediario no ofrece apenas mejoras para las solicitudes cuya respuesta deba buscarse en Internet. Debe exceptuarse el caso, en que las respuestas almacenadas en el intermediario permitan que el sistema deje de estar en saturación.

- El nivel de mejora, como era de prever, esta directamente relacionado con el tamaño del sistema de caches y proxies.

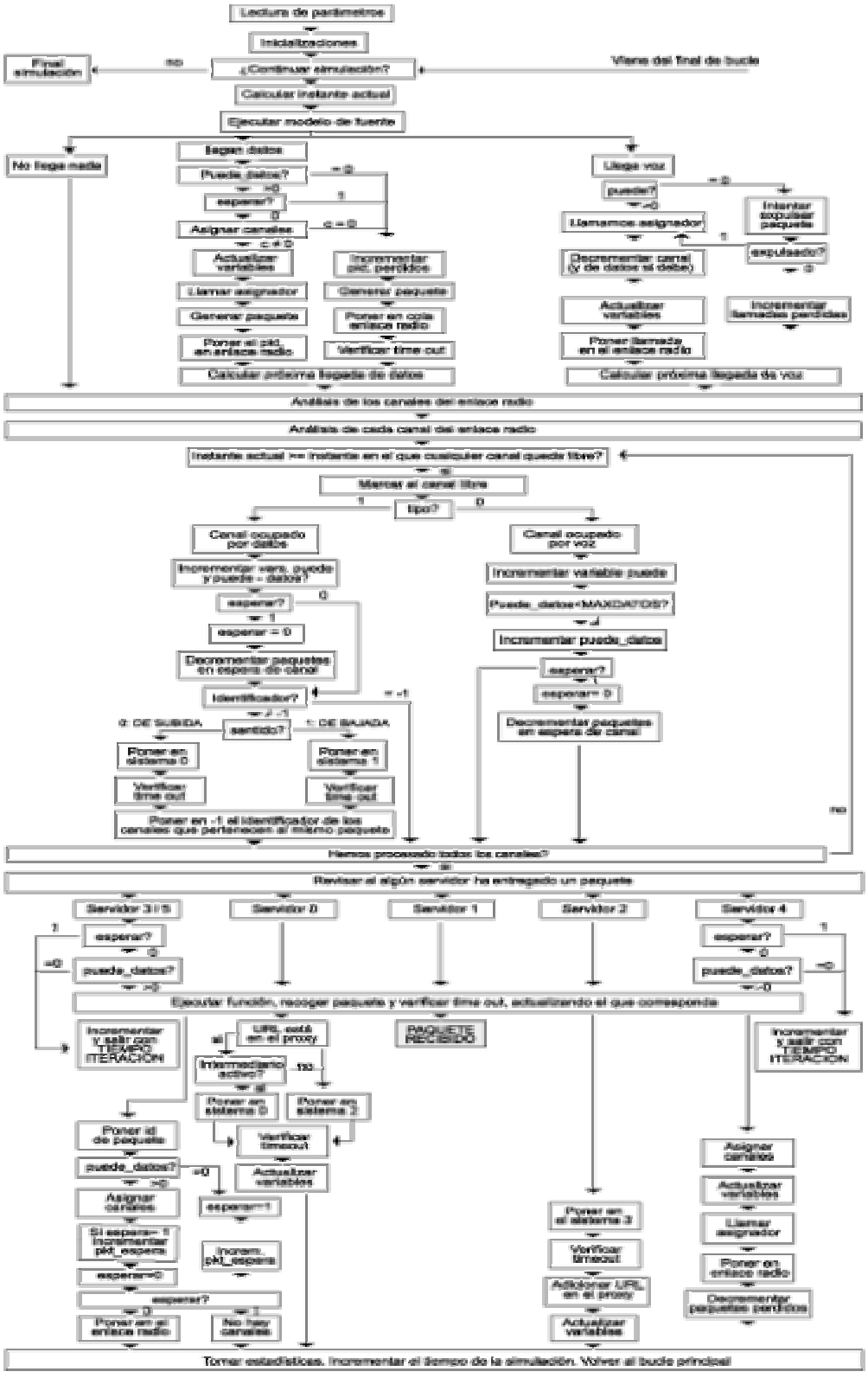


Fig.5.2 Esquema funcional del simulador.

5.5 Conclusiones.

En vista de los resultados obtenidos en las simulaciones que hemos llevado a término, podemos decir que la introducción del Intermediario en las proximidades de la pasarela WAP en el modelo utilizado actualmente es beneficiosa para la mejora general del modelo. Estudiando los resultados podemos ver que la eficiencia general del modelo aumenta considerablemente, es decir, hay muchas más peticiones que encuentran la página solicitada y que llegan exitosamente al usuario final, con lo cual aumenta el grado de satisfacción del usuario hacia este servicio, puesto que tendrá la certeza que la mayoría de las peticiones que haga serán tratadas y que al final obtendrá aquella información que había pedido.

Esta mejora en la eficiencia del sistema no sólo es en el volumen de tráfico tratado correctamente, sino que también se reduce de forma importante el tiempo que pasa desde que el usuario hace la petición hasta que le llega la página (retraso medio). La introducción del Intermediario hace posible que la petición se trate de manera inmediata si la página se encuentra en el proxy, de modo que la solicitud puede utilizar los mismos canales que ha utilizado la petición para llegar al proxy. De esta manera, se reduce de forma considerable la probabilidad de no encontrar ningún canal libre en el momento de poner la página en el enlace radioeléctrico en sentido de bajada. No obstante, para las peticiones que se han de dirigir a Internet para encontrar la página tardan un tiempo considerable entre llegar al servidor de Internet y que la página encontrada llegue al proxy, de modo que las condiciones de contorno hayan podido cambiar totalmente desde el instante que se realizó la petición, y posiblemente ahora no haya ningún canal libre en el enlace radio para poder enviar esta página al usuario.

El retraso medio de las páginas encontradas en Internet no mejora de forma sustancial, lo que sí mejora es el número de páginas servidas por el sistema que trata las páginas provenientes de Internet, puesto que merced a la introducción del proxy está mucho menos congestionado.

En resumen, la introducción del Intermediario en las proximidades de la pasarela WAP contribuye a reducir el tráfico por Internet, a aumentar la eficiencia de las búsquedas de páginas usando WAP y a reducir el tiempo medio de espera del usuario desde que envía una petición hasta que obtiene la información deseada, con lo cual la sensación subjetiva que le queda después de utilizar WAP es más gratificante que la que tenía con el modelo actual (sin Intermediario), puesto que no sólo recibe la mayoría de la información que ha pedido, sino que además la obtiene de forma más rápida y se convierta en un usuario habitual de este servicio, motivo principal al que van dirigidos la mayoría de los estudios actuales y uno de los motivos que impulsaron la realización de este trabajo.

Capítulo 6

Mecanismos complementarios para la entrega de información: Implantación del Intermediario.

6.1 Introducción.

Una vez realizado el diseño del Intermediario, simulado su comportamiento, y verificadas sus prestaciones, se inició la fase de desarrollo e implantación. Para ello se debieron elegir las herramientas y la plataforma software y hardware más adecuadas.

Como se expuso en el capítulo 4, el Intermediario consta de varios sistemas que se complementan. El primer requisito en el desarrollo del intermediario fue la portabilidad para poder integrarlo en cualquier plataforma. El segundo fue que las herramientas software utilizadas fuesen de libre distribución. Estas dos premisas nos hicieron decantar por el lenguaje de programación Java (aprovechando su característica multitarea), el uso de Aglets (soportados por Java) y la base de datos PostgreSQL, sobre entorno operativo Linux [LINU 01].

Las funciones de búsqueda pretenden optimizar el proceso de obtención de información y para ello necesitan componentes especializados como los agentes inteligentes, que pueden ofrecernos mayores y mejores funcionalidades en nuestro entorno de trabajo. El intermediario delegará las tareas de búsqueda a un sistema de agentes inteligentes, que conseguirá los resultados de forma segura, optimizando el tiempo de respuesta y facilitando la utilización de los servicios. Este sistema de agentes debe gestionar la transacción entre el cliente y los proveedores y para ello, realizará tareas tales como: buscar, aconsejar, contactar, comparar o filtrar.

El uso de estos mecanismos que posibilitan el trabajo cooperativo de múltiples agentes, facilita, además, tareas puntuales como por ejemplo, detección de proximidad geográfica con un centro de servicios, zonificación de la información o filtrado y tratamiento previo de la información que llega al equipo móvil. El intermediario tiene unas funciones de almacenamiento intermedio con las que se pretende agilizar el acceso de los usuarios a los contenidos de la Web, a través del mantenimiento de un caché de documentos. La información obtenida mediante de los mecanismos de búsqueda inteligentes es entregada a los clientes, pero también es clasificada y recogida en un almacén intermedio de documentos común a todos los usuarios.

El intermediario actuará como un servidor-proxy y atenderá todas las peticiones de los usuarios. Al recibir estas solicitudes, deberá comprobar en cada caso si una versión actualizada del documento se encuentra disponible en el caché y si es así, entregará al cliente esta

información guardada en memoria. En este caso, la respuesta del intermediario es mucho más rápida porque no requiere delegar las tareas de búsqueda a otros sistemas.

Sin embargo, si el documento no está disponible, la petición será enviada directamente a los mecanismos de búsqueda con agentes inteligentes. Respecto a la entrega de contenidos, de acuerdo con la consultora Jupiter Media Metrix, el servicio más deseado por los usuarios es el correo electrónico móvil. Por otra parte, The Yankee Group Europe en Londres apuesta por el servicio de mensajería SMS. Por consiguiente, la entrega de contenidos a los clientes se realizará en estas tres modalidades: WML/Script, e-mail y SMS.

En caso de desconexión eventual entre el intermediario y el resto de la red TCP/IP, el subsistema de caché es el único responsable de la entrega de documentos. En esta situación, el intermediario trabajará en modo de desconexión y sólo será posible el acceso y la distribución de contenidos previamente almacenados en memoria.

La personalización de servicios se consigue mediante la gestión de la relación con el cliente basándose en el perfil del usuario, en su actualización dinámica y en el análisis de sus necesidades.

El perfil del usuario no es más que el registro de información sobre un cliente en una base de datos, que incluye, principalmente, los hábitos y preferencias del usuario. Este perfil clasifica al cliente vinculándolo a ciertas listas de distribución y se actualiza dinámicamente cada vez que el usuario realiza una petición de servicio. Esto significa mejoras en la usabilidad, porque permite extender las posibilidades del equipo móvil y facilita su utilización.

En la modalidad de servicio push, el perfil es utilizado para determinar a que lista de distribución pertenece cada cliente. Los contenidos serán entregados adecuadamente a los componentes de cada lista de distribución con una frecuencia concretada por cada uno de ellos.

En cambio, cuando se realiza la búsqueda de información bajo pedido, el perfil del usuario ayudará a determinar de entre una amplia variedad de resultados, los más adecuados para ser entregados al cliente según sus necesidades.

Es conveniente que el intermediario almacene los certificados de autenticación de cada cliente, constituyendo un repositorio de certificados. Esta posibilidad permite a los clientes suministrar una URL donde encontrar los certificados, evitando así la obligación de mantenerlos en el dispositivo móvil. Todo esto forma parte de la evolución de los estándares, que tiende a establecer compatibilidad entre TLS y WTLS.

A continuación se detalla en primer lugar la implementación del sistema de búsqueda multiagente, luego el sistema de almacenamiento intermedio con cache y proxy; en tercer lugar, la personalización de servicios y finalmente el repositorio de certificados.

6.2 Implementación del Sistema de búsqueda multiagente.

En este apartado, tratamos de una forma detallada la implementación del sistema de búsqueda de información por Internet. El sistema de búsqueda lo constituye un sistema multi agente basado en tecnología Java y en Aglets [IBM 97]. Estudiamos su arquitectura, descripción funcional, conexión con la base de datos y detalles de la implantación.

6.2.1 Servidor de Aglets Tahiti.

El servidor de aglets Tahiti nos ofrece un entorno para ejecutar y visualizar los aglets que están ejecutándose. La arquitectura Aglets consta de dos capas, con sus correspondientes APIs que definen las interfaces para acceder a sus funciones, y son las siguientes:

- Aglet Runtime Layer o Capa de ejecución de Aglets con el API Aglet.
- Communication Layer o Capa de comunicaciones con el API Communication.

La Capa de Ejecución de Aglets. Define el comportamiento de los componentes del API: aglet, delegado del aglet, contexto, mensajes, etc. Proporciona las funciones fundamentales a los agentes para ser creados, enviados, administrados, y gestionados.

El núcleo de esta capa está compuesto por el Administrador de Seguridad, el Administrador de Cache y el Administrador de Persistencia. Proporciona los mecanismos fundamentales para la ejecución de aglet, como son: serialización y transferencia de clases, carga y transferencia de clases remotas o la limpieza de la memoria del servidor (Garbage Collector).

El Administrador de Seguridad es el responsable de proteger a los servidores y a los aglets de las entidades maliciosas. Intercepta todas las peticiones necesitadas de seguridad y chequea si el remitente tiene permiso para realizar la operación. El Administrador de Cache es el responsable de mantener el código utilizado por el agente. Por último, el Administrador de Persistencia es el responsable de guardar los agentes serializados, almacenando tanto su código como su estado en un método persistente como podría ser un disco duro.

La Capa de Comunicación. Es la responsable para la transmisión de un agente serializado al destino y para la recepción de éste. También soporta la comunicación entre agentes mediante el paso de mensajes. La fig. 6.1 muestra las capas de la arquitectura de Aglets

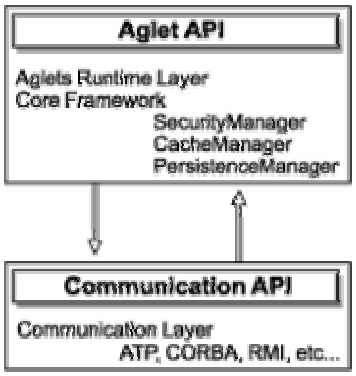


Figura 6.1 Arquitectura Aglet

Utilizamos el API de comunicaciones para transmitir, rastrear y administrar los agentes. Aglets utiliza el ATP (Agent Transfer Protocol) como implementación por defecto de la capa de comunicación. Protocolo de nivel de aplicación, modelado en el protocolo HTTP, para la transmisión de agentes móviles. Para habilitar la comunicación remota entre agentes, ATP soporta también el intercambio de mensajes. La figura 6.2 esquetamiza la comunicación entre agentes mediante el ATP.

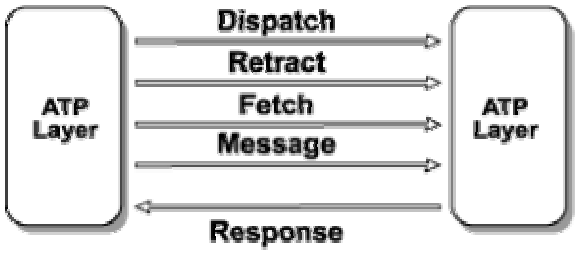


Fig. 6.2 Agent Transfer Protocol.

6.2.2 Arquitectura de sistema multiagente.

Una vez mencionadas las herramientas software utilizadas e introducido el servidor de Aglets Tahiti, podemos revisar la arquitectura del sistema multiagente del intermediario. En la figura 6.3 se esquematiza la arquitectura interna del sistema multiagente. Se pueden identificar los siguientes elementos:

Sistema operativo. El sistema operativo nos proporciona el entorno sobre el que se ejecutarán nuestras aplicaciones. La máquina virtual de Java y sobre esta el servidor de aglets, la base de datos y el socket servidor que espera peticiones de servicios.

6.2.2.1 Servidor de Aglets Tahiti.

El servidor de aglets permite la ejecución de los agentes dentro del intermediario, además de aportar las herramientas necesarias para la transferencia de agentes y clases. Para el correcto funcionamiento necesitamos el JDK (Java Development Kit) que provee de una máquina virtual sobre la que ejecutar el código generado.

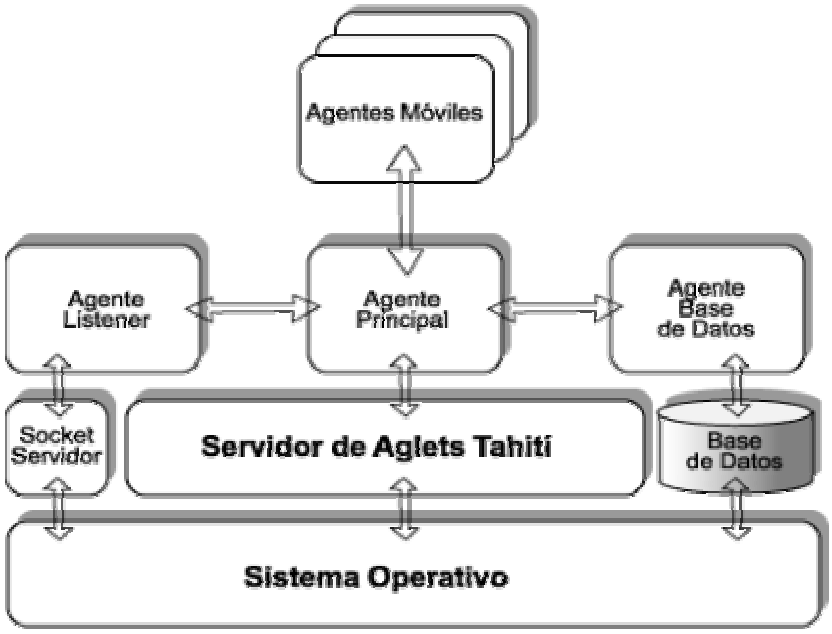


Fig. 6.3 Servidor de agentes Tahiti.

Base de Datos. Se utiliza la base de datos PostgreSQL. En ella se han registrado las tablas con la información de los servidores destino de cada uno de los agentes móviles, la clase relacionada a cada agente, además de una relación de las peticiones referentes al servicio de tiempo.

Socket Servidor. Las comunicaciones del servidor de agentes con el módulo encargado de recoger las peticiones del usuario se realiza mediante un socket. La aplicación mantiene un socket servidor abierto escuchando por el puerto 1500. Cuando se recibe una petición por parte del usuario, el módulo encargado de recibirlas, establece una comunicación con este socket y transmitirá la petición al intermediario. La utilización de sockets permite que la petición se realice desde cualquier ordenador o servidor.

Agente Principal. Es el agente más importante de nuestra plataforma. En este agente se centra la administración del resto de los agentes y la comunicación con el Agente de Base de Datos. Al participar de alguna manera en todas las tareas de la plataforma, se podría convertir en el cuello de botella. Para evitarlo podemos configurar, de igual forma que el Agente Listener, para que sea clonado cuando sea necesario.

Agente Listener. Se dispone de un Agente Listener encargado de escuchar por el puerto abierto a cualquier petición. Una vez recibida la petición crea un nuevo socket y delega la tarea de procesar la petición al Agente Principal. El Agente Listener volverá a su tarea de esperar nuevas peticiones. La programación realizada permite tener más de un Agente Listener, escuchando cada uno de ellos por un puerto diferente, con lo que se mejora la disponibilidad del intermediario. Además se puede clonar los agentes y ampliar automáticamente el número de Agentes Listeners según las necesidades.

Agente Base de Datos. El acceso a cualquier tipo de información interna por parte de los agentes, está centralizado en el Agente Base de Datos. Este agente posee la información necesaria en cuanto a localización de la base de datos, y dispone del controlador necesario para acceder a la información almacenada. Será el encargado de leer la información de los servidores destino de cada uno de los agentes móviles y pasarla al Agente Principal.

Agentes Móviles. Cada uno de los Agentes Móviles incluidos en la plataforma está ligado a un servicio. Por ejemplo, para el servicio de consulta de tiempo tenemos asociado un aglet específico. Agente Móvil con el conocimiento necesario para ir al servidor indicado, recoger la información y volver al servidor de agentes, para una vez allí, entregarle la información obtenida al Agente Principal. Cada uno de los aglets se comunica con el Agente Principal mediante el paso de mensajes propio de la plataforma Aglets. Así, definimos un tipo de mensaje para cada una de las posibles necesidades que puede tener cualquiera de los agentes.

6.2.3 Agentes Implantados.

En este punto explicaremos los agentes desarrollados, la funcionalidad y características de cada uno de ellos. En la siguiente figura 6.4 podemos ver todos los agentes desarrollados y la comunicación que existe entre cada uno de ellos.

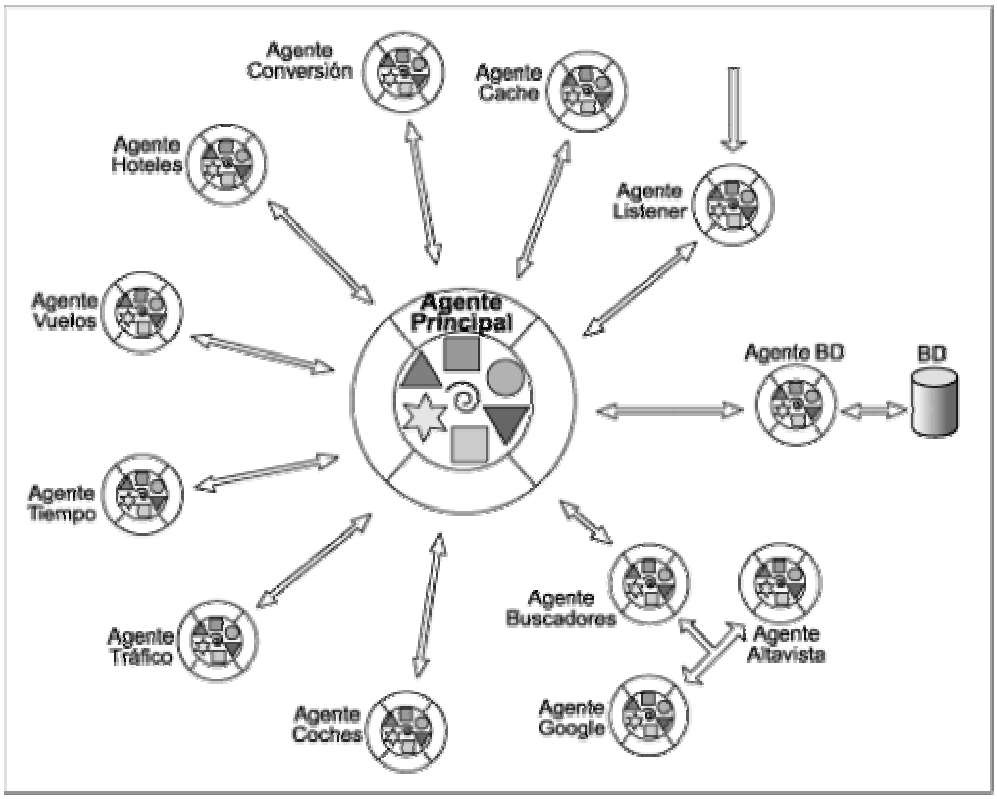


Fig. 6.4 Agentes desarrollados.

Como podemos ver, el Agente Principal gestiona en forma centralizada al grupo de agentes y la comunicación con cada uno de ellos. A continuación revisamos brevemente el funcionamiento de cada agente y sus características.

6.2.3.1 Agente Listener.

Agente estático encargado de recibir las peticiones. El Agente Listener mantiene un socket servidor abierto a la espera de recibir una petición de servicio. Utiliza el puerto 1500 como puerto de conexión. La conexión vía socket se realiza contra un puerto determinado en una dirección IP concreta. Una vez recibe una petición de conexión, crea un socket cliente, cuya administración delega al Agente Principal. A partir de este momento, el Agente Principal se encargará de dar respuesta a la petición. El Agente Listener continuará su tarea de escuchar una nueva petición por el puerto de conexión.

6.2.3.2 Agente Principal.

Agente estático donde centralizamos el funcionamiento del intermediario. Su funcionamiento es el siguiente:

- El Agente Principal recibe la referencia a un socket abierto.
- Obtiene una petición.
- Clasifica las peticiones entre las conocidas por el Intermediario.
- Se comunica con el Agente Base de Datos y le pide la clase asociada a la petición que ha recibido. El Agente Base de Datos a su vez abre una conexión con la base de datos local, forma la sentencia SQL de consulta y obtiene el resultado que lo entrega al Agente Principal.
- A partir del nombre de la clase, el Agente Principal crea al agente móvil específico para un servicio determinado.

- Consulta con el Agente Base de Datos la dirección del servidor destino del agente móvil creado.
- Le comunica al agente móvil específico su tarea y destino. El agente móvil migra al servidor destino, donde realiza su tarea, recoge los resultados y vuelve al intermediario.
- El Agente Principal recoge los resultados y los entrega por el socket que mantenía abierto desde el principio de la petición.
- Finaliza su ejecución a la espera de una nueva petición.

6.2.3.3 Agente Base de Datos.

Agente estático encargado de escribir y leer de la base de datos del intermediario. Recibirá peticiones las siguientes peticiones de lectura:

- Nombre de clase asociada a una petición.
- Servidor destino de una petición determinada.
- Información contenida en el cache.

Entre otras peticiones de escritura, recibirá la información sobre la previsión del tiempo. Esta información será reutilizable por el sistema, debe mantener un registro de si disponemos o no de la información del tiempo de un punto geográfico concreto.

El Agente Base de Datos, utiliza el controlador JDBC de Java correspondiente a nuestra base de datos (PostgreSQL). La comunicación con el Agente Principal la realiza mediante el intercambio de mensajes.

6.2.3.4 Agente Cache.

Este agente estático se encarga de administrar la información almacenada en el cache. Como hemos comentado antes, guardamos la información que puede ser reutilizable tal como la previsión del tiempo realizada para distintas ciudades con vistas a su posterior utilización. Esta información la guardamos en un sistema de archivos. El agente se encargará de limitar el crecimiento del caché, eliminando los archivos caducados. Esto es, si recibimos una petición que será almacenada, el agente busca peticiones para la misma ciudad en fechas anteriores y, en caso de existir, las elimina. Para facilitar la tarea al Agente Cache, hemos creado una tabla donde guardamos un registro de la información disponible referente al tiempo. Entonces, el Agente únicamente deberá solicitar esta información al Agente Base de Datos.

6.2.3.5 Agentes Móviles.

Los agentes móviles de búsqueda necesitan del AWB (Aglet WorkBench de IBM). AWB es la base para el funcionamiento, comunicación y movilidad de los aglets. Para que el agente sea capaz de ir a cualquier proveedor de contenidos, debe disponer de un servidor de aglets. Esta característica, necesaria para componer el sistema global de agentes, no se cumple en la actualidad. Por tanto, deberemos buscar alternativas para dar respuesta a los servicios que ofrece el intermediario a los usuarios móviles. Lo que se ha hecho es buscar en Internet portales y servidores adecuados que poder utilizar para obtener la información.

Por ejemplo: para obtener la información del tráfico, el aglet realiza una conexión HTTP al servidor de la Dirección General de Tráfico y realiza la consulta automáticamente guiado por la petición del usuario. Luego, filtra la información y se la entrega al Agente Principal, que a su vez la entrega al usuario.

Para comprobar la correcta movilidad de los agentes, hemos utilizado dos servidores de agentes dispuestos en dos servidores distintos. Así, cada uno de los agentes es creado en un servidor y enviado al otro para realizar su tarea. Una vez obtenidos los resultados, vuelve al servidor origen y entrega los resultados al Agente Principal. Cada servidor de agentes se

identifica por el nombre de la máquina donde está ejecutándose y un puerto. El puerto por defecto es el 4434, pero podemos utilizar cualquier puerto libre del sistema. Entonces, para que un agente pueda migrar a otro servidor, deberemos indicarle dirección del servidor más el puerto.

A continuación explicaremos el funcionamiento y peculiaridades de los distintos agentes móviles implementados.

Agente Buscador. Realiza búsquedas sirviéndose de buscadores disponibles en Internet. A partir de la petición que el Agente Principal le comunica, crea instancias de dos agentes específicos de buscadores: Agente Google y Agente Altavista. Cada uno de ellos tiene el conocimiento de cómo acceder a estos motores de búsqueda, lanzar la búsqueda y recoger los resultados. Además, cada uno de ellos extrae los enlaces y sus descripciones contenidos en una página HTML. El Agente Buscador recoge los resultados y elimina la información redundante.

Agentes Altavista y Google. Delegan las búsquedas sirviéndose de los buscadores Altavista y Google respectivamente. Cada agente conoce la dirección URL del buscador correspondiente, los argumentos necesarios para delegar una búsqueda y recoger los resultados en una página HTML. Se filtra la página utilizando marcas y patrones, extrae las URLs referenciadas y sus respectivas descripciones.

Agente Vuelos. Agente móvil encargado de realizar las consultas de disponibilidad de vuelos. Para realizar la consulta necesitará información acerca del origen y destino, además de las fechas y horas. Accederá al servidor de Amadeus, utilizado por las agencias de viajes para mirar la disponibilidad de vuelos y realizar las reservas. El Agente de Vuelos, es similar a los Agentes Coches, Hotel y Conversión de moneda.

La información de disponibilidad de vuelos que se puede obtener de Amadeus.net está orientada a usuarios humanos. Cada vez que se conecta a su servidor, se obtiene una clave de sesión que identifica al usuario y le permite realizar consultas.

Al realizar los aglets el acceso de forma automática, se encontró el problema de los identificadores temporales. El problema se superó de la siguiente manera:

- El aglet que debe acceder a cualquier información haciendo uso de Amadeus.net, se conecta a la página donde podemos consultar la disponibilidad.
- Realiza un rastreo de la misma utilizando un patrón de la referencia del identificador de sesión.
- Una vez lo obtiene, puede formar la consulta basándose en las necesidades del usuario e incluir en ella la clave de sesión obtenida.
- De esta forma, para cada acceso nuevo a los servidores de Amadeus obtenemos una nueva clave de sesión que nos habilita para utilizar sus servicios.
- La información de la disponibilidad está dispuesta en una página HTML. Contiene información de las fechas y horarios, así como las compañías.

Se aprovecha el conocimiento adquirido en la utilización de los servicios ofrecidos por Amadeus.net para acceder a los servicios de disponibilidad de coches de alquiler y hoteles.

Agente Coches. Agente móvil encargado de realizar las consultas de disponibilidad de coches de alquiler. Para la consulta nos basamos en las necesidades del usuario en cuanto a fechas, ciudades y modelo del coche. Accedemos a los servidores y recogemos los resultados en una página HTML donde encontramos información acerca de la casa de alquiler de coches y la disponibilidad por clase de coche.

Agente Conversor. Agente móvil encargado de realizar conversiones de moneda entre las principales del mundo. Para la consulta necesitamos los datos del valor a convertir y el origen y destino de la moneda. Este Agente nace para dar respuesta a un servicio de valor añadido. Servicio del que podemos disponer en los servidores de Amadeus.net.

Agente Hoteles. Agente móvil encargado de mirar la disponibilidad de hoteles en una ciudad determinada y en unas fechas concretas. Para esto, necesita los valores de las fechas de entrada y salida, además de la ciudad. Devuelve una página HTML con una relación de hoteles disponibles, incluyendo información de categoría, precio, dirección y teléfono. El Agente Hoteles también se vale de los servicios ofrecidos por Amadeus.net. A partir de las fechas solicitadas por el usuario, la ciudad elegida y la categoría demandada, el agente forma la URL correspondiente y lanza la petición. Obtiene los resultados en forma de página HTML. De igual forma que el Agente Vuelos, este agente obtendrá la clase sesión correspondiente para acceder a los servicios de Amadeus.net.

Agente Tiempo. Agente móvil encargado de obtener la información referente al tiempo en la fecha de la petición y la previsión para los cuatro días siguientes. A partir de la ciudad donde se encuentre el usuario u otra cualquiera, el agente accede al servicio de previsión del tiempo de Yahoo obteniendo una página HTML con los gráficos correspondientes. Una vez obtiene esta página, realiza un filtrado de la misma buscando una serie de patrones e identifica la previsión de un día en forma de una palabra: soleado, cubierto, chubascos, etc.

Creemos necesario realizar el filtrado obteniendo un vector de cinco palabras referente a la previsión del mismo día y los cuatro siguientes. Esto nos permitirá formar un mensaje SMS para enviar al usuario que ha realizado la petición. Éste es un claro ejemplo de la utilidad de la entrega de la información de forma asíncrona. El usuario puede realizar la petición de previsión del tiempo en forma de mensaje (tiempo:Barcelona) y desconectarse. El intermediario, por medio del sistema de agentes obtiene la información, forma el mensaje SMS y se lo envía al usuario.

Además, nuestro sistema dispone de un caché referente a las peticiones del tiempo. De este modo, si recibimos una petición sobre la previsión en la ciudad de Barcelona, miraremos si hemos recibido otra petición igual durante el mismo día. Si es así, devolveremos la información que ya fue obtenida en su momento sin necesidad de lanzar otra búsqueda. Con la utilización del caché reducimos tanto el tiempo de respuesta como los recursos del sistema utilizados.

Agente Tráfico. Agente móvil encargado de obtener la información referente al estado de las carreteras en una provincia o comunidad autónoma determinada. Puede discriminar entre el tipo de anomalía, obteniendo información de la situación de las carreteras debida a obras, atascos, tiempo o cualquier otra razón. Este agente está especializado en el acceso a los servidores de la Dirección General de Tráfico. Obtiene los argumentos necesarios para lanzar la petición, como son si realizamos la búsqueda por provincia o comunidad autónoma y si mostramos todos los problemas o sólo algunos.

6.2.4 Implementación y aspectos de seguridad en el sistema de búsqueda multiagente.

La implementación del sistema multiagente se resume en los siguientes pasos:

- Instalación del JDK 1.1.8. Proporciona el API de java, además de la JVM.
- Instalación y configuración del servidor de aglets Tahiti.
- Diseño y desarrollo de los aglets y las clases necesarias.
- Instalación y configuración de la base de datos PostgreSQL.
- Instalación del controlador para la base de datos.
- Configuración del Aglet PostgreSQL para la conexión del servidor de agentes con la base de datos.
- Instalación de un segundo servidor de aglets en otro PC.
- Pruebas de funcionamiento y movilidad de todos los aglets.

En la implementación del intermediario y por tratarse de un prototipo no se ha profundizado el estudio de las medidas de seguridad de Java y Aglets. Disponemos sin embargo, de funciones criptográficas implementadas en Java e incorporadas en JDK a partir de la versión 1.2. En cuanto a la seguridad del servidor de agentes, podemos definir los permisos referentes a todos los elementos de Aglets. Definimos los permisos referentes al acceso a los recursos del sistema, conexiones HTTP o ATP, cifrado de los mensajes, control del acceso a los aglets, control de acceso al contexto creado, entre otros. En la implementación realizada, accedemos a los servidores remotos utilizando el protocolo de transporte HTTP, aunque podemos optar por utilizar un protocolo para conexiones seguro como es el HTTPS o TLS.

6.2.5 Funcionamiento del Sistema de búsqueda multiagente.

El funcionamiento del sistema de búsqueda se esquematiza en la figura 6.5 y a continuación se describe paso a paso.

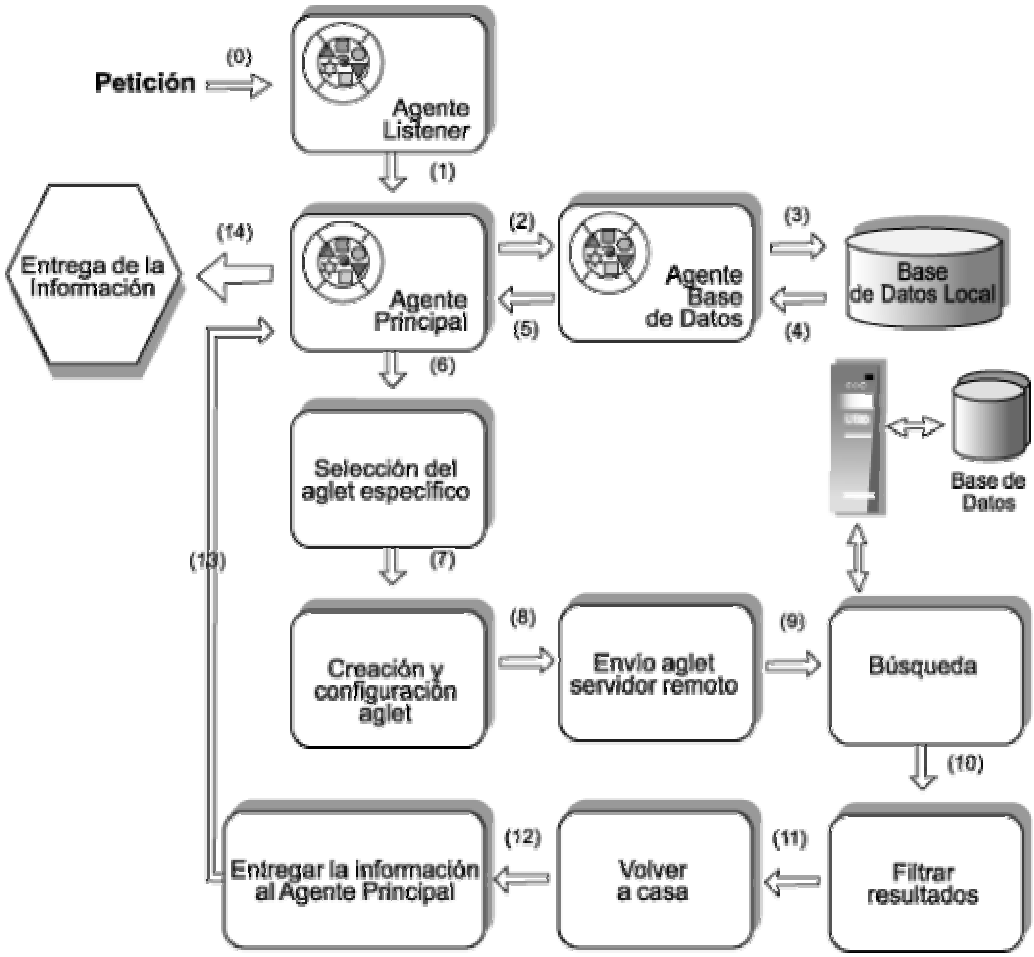


Fig. 6.5 Esquema de funcionamiento del Intermediario.

0. El Intermediario recibe una petición por medio del Agente Listener. La recibe por el puerto 1500, donde mantiene un socket servidor abierto.

1. Una vez recibida la petición, el Agente Listener crea un socket cliente con la conexión establecida con la fuente de la petición y delega el trabajo de atenderla al Agente Principal. El Agente Listener sigue con su tarea de escuchar por el puerto 1500.

2. El Agente Principal analiza la petición para extraer el servicio requerido. Crea el Agente Base de Datos y le pide la información sobre el nombre de la clase asociada al servicio (AgletBuscador, AgletTiempo, AgletVuelos, etc.) y la dirección del servidor destino de éste. A partir de esta información crea al agente móvil específico.

3, 4. El Agente Base de Datos se conecta con la base de datos local (PostgreSQL). Extrae la información referente al servicio demandado.

5. Una vez obtenida la información, el Agente Base de Datos entrega la información al Agente Principal mediante el intercambio de mensajes.

6, 7, 8. A partir de la información obtenida del Agente Base de Datos, el Agente Principal crea el aglet específico para el servicio que ha sido solicitado (Información de carreteras, previsión del tiempo, disponibilidad de vuelos, disponibilidad de hoteles, disponibilidad de coches de alquiler, servicio de conversión de moneda o buscador). Al crear el aglet, le pasa como

parámetro la dirección del servidor destino y los argumentos necesarios para realizar su tarea. El aglet migra a su servidor destino.

9, 10. El aglet llega al servidor remoto, realiza su tarea, dedicada a obtener la información para la que ha sido enviado y, si es oportuno, filtra los resultados.

11. Una vez ha obtenido la información, el aglet se prepara para volver a casa.

12,13. Una vez en casa, entrega la información al Agente Principal. El aglet será desactivado hasta que sus servicios sean requeridos.

14. El Agente Principal, que mantenía el socket abierto con la fuente de la petición, entrega por el socket la información obtenida. Cierra las comunicaciones, a la espera de una nueva petición.

6.3 Sistema de cache y proxy.

El almacenamiento intermedio se implementa construyendo una base de datos en la que se recogen los resultados de las búsquedas en Internet. Para acceder a la información almacenada se utiliza el lenguaje SQL y la base de datos se indexa con una palabra clave que nos permita realizar estos accesos de forma rápida y eficiente. De manera similar, en la base de datos del perfil del usuario se almacena información personal y profesional del cliente y sus gustos y preferencias. Estos datos son actualizados automáticamente cada vez que el usuario interacciona con el sistema.

A continuación nos referimos a los pasos mostrados en el diagrama de flujo de la figura 6.6.

1. Cuando un usuario realiza una petición para obtener un documento, el micro-navegador hace llegar la solicitud al intermediario a través del gateway WAP. Si la petición es aceptada, los mecanismos de gestión del perfil del usuario actualizan el perfil con los datos proporcionados en la solicitud del cliente. Si la petición se formulara incorrectamente y no fuera aceptada por el intermediario, se pide al usuario que la formule de nuevo.

2. El intermediario realiza la consulta en la base de datos de documentos en busca de la información solicitada.

3. Si se encuentra una copia reciente en el caché, el intermediario averigua en el perfil del usuario cual es el formato de entrega adecuado, y prepara y envía los contenidos al cliente. Si no se encuentra la información deseada, la petición se traslada a los mecanismos de búsqueda en Internet.

4. El sistema multiagente se encarga de la búsqueda y devuelve los resultados al intermediario. Los parámetros de la búsqueda se pueden completar con algunos datos del perfil del usuario, que permitirán personalizar y concretar los contenidos entregados.

5. Si la búsqueda resulta fallida, se envía al usuario un mensaje para que redefina su petición y se inicia un nuevo proceso. Cuando se obtiene la información deseada, se guarda una copia en caché y se prepara la entrega de contenidos en el formato deseado por el cliente.

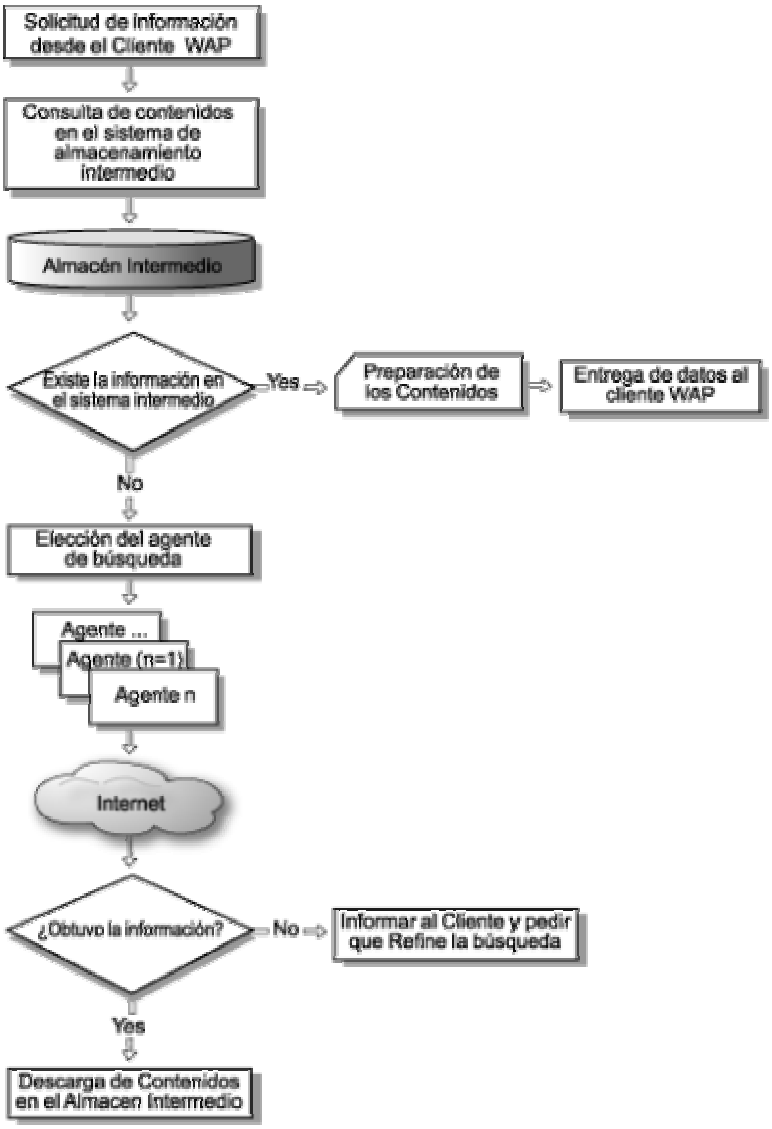


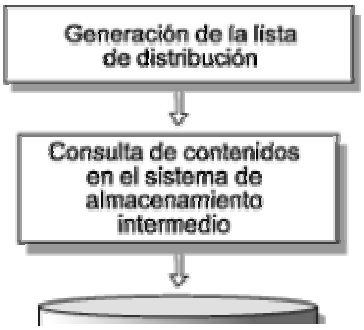
Fig. 6.6 Diagrama de flujo en modalidad Pull

Cuando se trata de la entrega de contenidos en modalidad push, el método que se sigue es ligeramente diferente, como se puede ver en la figura 6.7.

1. Con la información de que dispone el intermediario en los perfiles de usuario, se agrupan los clientes según sus intereses y preferencias, formando listas de distribución de contenidos.

2. El almacén intermedio de documentos es revisado por el intermediario en busca de información que pueda ser interesante para los usuarios.

3. Con los resultados obtenidos, se preparan los documentos para ser entregados a cada componente de las listas de distribución.


Generación de la lista de distribución
Consulta de contenidos en el sistema de almacenamiento intermedio