

MACSIM: DISEÑO, IMPLEMENTACIÓN Y LIBERACIÓN

Máster Oficial en Software Libre

Santiago Cambero

Autor

Gregorio Robles

Director PFM

20/01/2008

Copyright (c) 2008 SANTIAGO CAMBERO

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

RESUMEN

En el presente proyecto se expone el diseño, la implementación y la liberación de una aplicación de software libre denominada MACsim. Esta herramienta permite simular el funcionamiento de los protocolos MAC (Medium Acces Control) de las redes de comunicaciones móviles de tercera generación. En concreto, se implementa el protocolo WISPER.

ÍNDICE

CAPÍTULO 1. INTRODUCCIÓN.....	9
1.1. Motivación.....	9
1.2. Objetivos.....	9
1.3. Metodología.....	10
CAPÍTULO 2. ESTUDIO DE VIABILIDAD.....	13
2.1. Tareas.....	13
2.2. Planificación.....	16
CAPÍTULO 3. FUNDAMENTOS TEÓRICOS.....	17
3.1. Introducción.....	17
3.2. Objetivos.....	17
3.3. Escenario.....	18
El enlace ascendente.....	18
El enlace descendente.....	20
3.4. Consideraciones previas.....	21
3.5. El módulo planificador de paquetes.....	22
Cálculo de prioridad.....	23
Procedimiento de acomodación de paquetes.....	25
Procedimiento de asignación de paquetes.....	27
3.6. Ejemplo.....	30
3.7 Tipos de tráfico.....	36
Voz.....	36
Vídeo VBR (Variable Bit Rate).....	39
Vídeo CBR (Constant Bit Rate).....	40
Audio CBR (Constant Bit Rate).....	41
Datos ABR (Available Bit Rate).....	41
Correo ABR (Available Bit Rate).....	42
CAPÍTULO 4. DISEÑO E IMPLEMENTACIÓN.....	43
4.1. Diseño.....	43
4.2. Implementación.....	44
CAPÍTULO 5. FUNCIONAMIENTO.....	47
5.1 Menú Setup.....	48
Logger.....	48
Statistics.....	48
Frame.....	49
Slot.....	50
Traffic.....	50

General.....	51
ABR Computer Data.....	51
ABR Email.....	52
CBR Audio.....	53
CBR Video.....	53
VBR Video.....	54
Voice.....	55
5.2 Menú Run.....	56
Run.....	57
Stop.....	57
5.3 Menú Help.....	57
About.....	57
CAPÍTULO 6. LIBERACIÓN.....	59
6.1. Elección de la licencia.....	59
6.2. Documentación del código fuente.....	59
6.3. Preparación del paquete MACsim_v1.0.tar.gz.....	59
6.4. Alta de usuario en SourceForge.....	60
6.5. Alta del proyecto en SourceForge.....	60
6.6. Liberación del paquete MACsim_v1.0.tar.gz.....	61
CAPÍTULO 7. CONCLUSIONES.....	63
REFERENCIAS BIBLIOGRÁFICAS.....	66
GNU Free Documentation License.....	68

CAPÍTULO 1. INTRODUCCIÓN

1.1. Motivación

El presente proyecto final de máster consiste en el desarrollo y liberación de una aplicación (MACsim) para la simulación de protocolos MAC (Medium Access Control) en sistemas CDMA (Code Division Multiple Access).

Las diferentes variantes de este tipo de acceso (entre los que destacan W-CDMA, UMTS-TDD y EV-DO) son los utilizados en los sistemas de comunicaciones móviles de tercera generación, tanto de telefonía como Internet. Debido a que en las nuevas redes de comunicaciones móviles adquieren especial relevancia la transmisión de tráfico multimedia que origina un consumo elevado de ancho de banda, los antiguos sistemas de acceso múltiple (TDMA – Time Division Multiple Access y FDMA - Frequency Division Time Access) han sido sustituidos por CDMA.

Por lo tanto, la problemática de la gestión de recursos (donde se engloban los protocolos MAC) en sistemas CDMA es un campo donde actualmente se realiza un gran trabajo de investigación por parte de universidades y empresas.

Los objetivos de este proyecto son desarrollar y liberar una aplicación que permita simular estos protocolos MAC. En concreto, la aplicación simula un único protocolo, denominado WISPER. Aunque el objetivo es que se convierta en una plataforma que soporte múltiples protocolos, con el fin de poder analizarlos y compararlos. Es por lo tanto, un proyecto totalmente abierto y ampliable.

Las motivaciones para la elección de este proyecto han sido dobles. Por un lado, como se ha comentado anteriormente, es un campo donde en la actualidad se realiza una importante labor de investigación. Constantemente, se proponen nuevos protocolos, así como modificaciones de los existentes. Por otro lado, aunque existen simuladores para estos protocolos, no hay una plataforma que englobe a varios de ellos para compararlos de una manera uniforme y homogénea.

Como consecuencia de este proyecto se pretende crear una aplicación de utilidad para la investigación y evaluación de protocolos MAC en el campo de las comunicaciones móviles de tercera generación.

1.2. Objetivos

El objetivo general de este proyecto es proporcionar una aplicación que simule un protocolo de acceso al medio MAC (denominado WISPER). La aplicación contará con una interfaz gráfica donde se definirán todos los parámetros de

simulación. Además, se podrá seguir la evolución de la simulación por medio de gráficas, tanto a nivel de tráfico generado como de rendimiento, paquetes perdidos y retardo medio. Finalmente, se generarán unas estadísticas para cada simulación, que se podrán exportar a formato CSV, XML o guardarse en una Base de Datos (MySQL). Debido a que en el ámbito del proyecto se realiza una importante labor de investigación y a la inexistencia de una herramienta de este tipo, se intentará involucrar a otros usuarios y desarrolladores. La finalidad es extender la aplicación y que soporte otros tipos de protocolos, tales DPRMA, CPRMA, DTDMA, MASCARA, MDPRMA-BB o WCDMA.

El objetivo particular del proyecto es desarrollar y liberar una aplicación siguiendo la metodología de desarrollo del software libre. En concreto, deberán realizarse las siguientes tareas sobre la aplicación:

- Elegir la licencia.
- Desarrollarla en un entorno de desarrollo libre.
- Documentarla, tanto a nivel de usuario como de desarrollador.
- Empaquetarla, con especial énfasis en simplificar su instalación.
- Hospedarla en alguno de los portales que alojan software libre, haciendo uso de las herramientas que proporcionan:
 - Control de versiones.
 - Sistemas de seguimiento de fallos.
 - Sección de descargas (software y documentación).
 - Página web.
 - Listas de correo.
- Localizar / Internacionalizar.
- Hacer publicidad del proyecto.

1.3. Metodología

Para alcanzar los objetivos especificados en el apartado anterior, se ha seguido la metodología que se describe a continuación, que también es aplicable a la organización de la presente memoria.

En un primer lugar se ha realizado la planificación del proyecto que se expone en el segundo capítulo. En base a los objetivos a alcanzar se han identificado una serie de tareas, tanto teóricas como prácticas, y se han planificado en función de la previsión temporal de cada una de ellas.

El siguiente paso ha sido el análisis del protocolo WISPER, que se presenta de forma teórica en el tercer capítulo. En primer lugar, se describen los objetivos del protocolo, que no son otros que maximizar el rendimiento y minimizar las pérdidas. Para maximizar el rendimiento, el protocolo planifica la transmisión de los distintos paquetes en función de la tasa de bits erróneos de cada tipo de tráfico (BER). Las pérdidas se minimizan mediante un esquema de prioridades en función del tiempo de expiración de los paquetes.

En este mismo capítulo, se describe el escenario de funcionamiento del protocolo, esto es, un sistema CDMA con un enlace ascendente y otro descendente. Se muestra la división en tramas y ranuras de ambos enlaces (con ranuras de control, petición y datos) y el decalado entre ambas.

En el siguiente apartado, se amplían las consideraciones en cuanto al parámetro BER para optimizar la asignación de recursos y el uso del tiempo de expiración para minimizar la pérdidas. Este último no es el único método posible, pero es el escogido por WISPER.

A continuación, se presenta el módulo planificador de paquetes y las expresiones para calcular la prioridad de los paquetes y el número de paquetes con permiso de transmisión. La prioridad debe ser directamente proporcional al número de paquetes a transmitir e inversamente proporcional a su tiempo de expiración. El número de paquetes con permiso de transmisión es directamente proporcional a la prioridad. También se describen la estructura de registros y grupos de paquetes asociados que monta el protocolo con las peticiones que recibe, así como los algoritmos de acomodación y asignación de paquetes. El primero de ellos, elige grupos de paquetes con la misma prioridad y calcula cuantos tienen permiso de transmisión en la próxima trama. El segundo algoritmo, asigna los paquetes en las ranuras, buscando siempre agrupar los que tienen un requerimiento BER similar. A continuación, se muestra un ejemplo práctico del funcionamiento del protocolo WISPER.

Para finalizar el análisis teórico se exponen los tipos de tráfico presentes en los sistemas de comunicaciones móviles de tercera generación. Estos tipos de tráfico son voz, vídeo VBR, vídeo CBR, audio CBR, datos ABR y correo ABR. En este punto, pueden comenzar a observarse las diferencias en la demanda de recursos que hay entre el tráfico de voz y datos y el tráfico multimedia.

El cuarto capítulo comienza con una breve descripción del simulador implementado. El lenguaje de programación utilizado es Java y el entorno de desarrollo es el IDE Eclipse sobre un sistema Debian GNU/Linux. Se describen los módulos del simulador, los parámetros de inicialización, la interfaz gráfica así como algunas recomendaciones a la hora de ejecutarlo, tales como deshabilitar la función de traza en las simulaciones de carga.

En el quinto capítulo se expone el funcionamiento de la aplicación desde que se recibe una petición de transmisión en la cola de peticiones, se pasa a la cola interna de registros, se calcula su prioridad y se asigna en una ranura mediante el procedimiento de asignación de paquetes. También se describen las distintas pantallas, menús y diálogos para configurar la aplicación, así como el significado de cada uno de los elementos.

En el sexto capítulo se expone la liberación de la aplicación. Para ello se ha dado de alta un proyecto en SourceForge y se han seguido una serie de recomendaciones tanto en el desarrollo, la instalación de la software y las utilidades necesarias para gestionar su liberación.

En el séptimo y último capítulo se ofrecen las conclusiones del presente trabajo,

tanto de la parte teórica como de la práctica, así como un análisis de los objetivos conseguidos total y parcialmente.

También se indican algunas posibles actualizaciones o extensiones del presente trabajo. Entre ellas, destacan la implementación de protocolos como MDPMA-BB o WCDMA.

Finalmente, se proponen algunas mejoras y recomendaciones de cara al usuario (interfaz gráfica, módulos exportadores, etc.) y a conseguir la atención de la comunidad.

CAPÍTULO 2. ESTUDIO DE VIABILIDAD

En este capítulo se identifican las tareas a realizar para la implementación y liberación de la aplicación MACsim. Seguidamente se presenta la planificación temporal inicial de dichas tareas.

2.1. Tareas

A continuación, se detallan cada una de las tareas a realizar en función de los objetivos marcados en el proyecto:

- Estudio teórico.

Debe estudiarse de forma teórica el protocolo WISPER, que en un primer momento es el único protocolo MAC que implementa la aplicación MACsim.

- Elección de la licencia.

El software a implementar debe ser libre. Entre las licencias de software libre, la más difundida es la GPL (GNU Public License), que permite la distribución binaria y del código fuente y realizar modificaciones sin restricciones o integrarlo en otras aplicaciones siempre que el código resultante tenga una licencia GPL o compatible. Por lo tanto, se adapta a las necesidades actuales y futuras de la aplicación MACsim. En concreto, la licencia elegida para el proyecto es la GPLv3, que aporta cláusulas de defensa contra las patentes de software, resuelve ambigüedades de la versión anterior y facilita su adaptación a otros países.

- Implementación (lenguaje y entorno de desarrollo).

Debido al poco tiempo disponible para realizar el proyecto y con el objetivo de llegar al mayor número de usuarios posibles (tanto de entornos GNU/Linux como Windows), se ha optado por desarrollar MACsim en el lenguaje de programación Java. Este lenguaje disminuye el rendimiento de las aplicaciones debido a la abstracción hardware que realiza. Por este motivo, es un lenguaje multiplataforma y conveniente para la implementación rápida de aplicaciones en entornos heterogéneos.

Actualmente, los dos principales entornos de desarrollo en Java de software libre son NetBeans (SUN) y Eclipse (Eclipse Foundation). Se ha optado por la segunda opción, ya que dispone de un editor visual para crear interfaces de usuarios, así como una serie de conectores para trabajar con XML, bases de datos, realizar pruebas unitarias con JUnit, desarrollar proyectos en C/C++ (entre otros lenguajes), etc. Otro aspecto a tener en cuenta es que su difusión es mayor que NetBeans. Por lo tanto, será más fácil involucrar a

desarrolladores si conocen el entorno de desarrollo original.

De todas formas, Eclipse únicamente proporciona el entorno de desarrollo, las herramientas de compilación, debug, documentación e intérprete pertenecen al JDK 1.6 de SUN.

- Documentación.

La documentación se divide en documentación para usuarios y desarrolladores. En el primer caso, se utilizará DocBook con el objetivo de generar la documentación final en formato pdf y html. Para documentar el software, se utilizará la herramienta javadoc del JDK.

- Instalación del paquete.

La aplicación se liberará en formato binario mediante un fichero '.tar.gz' para entornos GNU/Linux y un fichero '.zip' para entornos Windows. Adicionalmente, se distribuirán las fuentes que deberán construirse con la aplicación Ant. Sería también interesante disponer de distribuciones en alguno de los sistemas de paquetes más utilizados en GNU/Linux (DEB y RPM). Lógicamente, también habrá que generar documentación para la instalación.

- Hospedaje el proyecto.

El proyecto se alojará en SourceForge para poner la herramienta a disposición de la comunidad. Se ha elegido este sitio, a pesar de existir otras alternativas, porque en la actualidad es el que tiene una difusión más amplia, tanto en número de usuarios como proyectos registrados.

Los pasos necesarios y tareas a realizar en esta fase del proyecto son las siguientes:

- Registro del proyecto.

Debido a que el registro del proyecto no es automático (en general, se tardan varios días en recibir la aprobación) es conveniente realizarla dentro de las primeras semanas de la planificación.

- Configuración del CVS.

Eclipse dispone de un conector para acceder repositorios CVS remotos. Una vez configurado, desde el mismo entorno de desarrollo se podrá gestionar el sistema de control de versiones alojado en SourceForge.

- Sistemas de seguimiento de fallos.

El sistema de seguimiento de fallos de SourceForge se denomina Tracker. Permite la gestión de fallos, peticiones de soporte, peticiones de funcionalidad y parches.

- Listas de correo.

Las listas de correo en SourceForge emplean el gestor GNU Mailman. Hay que tener en cuenta que no se crean de forma automática, sino mediante una tarea programada en el cron. Sería conveniente disponer de una lista de correo para usuarios y otra para desarrolladores.

- Sección de descargas (software y documentación).

Es otra de las herramientas que ofrece SourceForge. En un primer momento, se crearán cuatro paquetes: uno para la distribución en GNU/Linux, otro para la versión en Windows, otro con los ficheros fuente y otro con la documentación. También sería interesante disponer de paquetes estándar, tales como DEB (para las distribuciones basadas en Debian) o RPM (para las basadas en RedHat o que han adoptado este tipo de gestor).

- Web.

En un primer momento, la web del proyecto será la página principal del proyecto en SourceForge. Antes de la liberación, se creará una web (dentro de SourceForge) con un aspecto más publicitario, de cara a dar una buena impresión a los visitantes del proyecto. Las secciones que debe contener la página son las siguientes: Proyecto, Listas de Correo, Noticias, Descargas, Desarrollo, Foros, Documentación, etc.

Exceptuando el registro del proyecto, que puede tardar varios días en aprobarse, el resto de tareas en SourceForge conviene realizarlas una vez se disponga de una versión funcional mínima de la herramienta. El objetivo es dar una sensación de continuidad en el desarrollo de la aplicación.

- Localizar / Internacionalizar.

Para facilitar una mayor difusión, el idioma de la aplicación, documentación, web, etc. será el inglés.

- Hacer publicidad del proyecto.

La publicidad del proyecto se realizará una vez este disponible la primera versión completa (con todos los módulos) de la herramienta. Para dar publicidad al proyecto podemos publicarlo en portales como <http://freshmeat.net>, <http://dmoz.org>, <http://linux.bankhacker.com>, <http://barrapunto.com>, <http://libertonia.escomposinux.org> o <http://slashdot.org>.

2.2. Planificación

Las tareas pueden dividirse en tareas de teóricas, de desarrollo y de liberación/difusión. Para una mayor claridad, la planificación de cada una de ellas se ha realizado por separado, aunque algunas de ellas se solapan en el tiempo.

Las tareas teóricas se resumen en la siguiente tabla:

Tabla 2.1 Tareas de teoría

TAREA	FECHA INICIO	FECHA FIN
Estudio del protocolo WISPER	07/11/2007	10/11/2007
Estudio de los tipos de tráfico presentes en el sistema (voz, datos, audio, vídeo)	11/11/2007	12/11/2007

Las tareas de desarrollo se resumen en la siguiente tabla:

Tabla 2.2 Tareas de desarrollo

TAREA	FECHA INICIO	FECHA FIN
Migración de los distintos tipos de tráfico (voz, datos, multimedia) a Java	13/11/2007	18/11/2007
Desarrollo del interfaz gráfico para configurar los distintos tipos de tráfico y las gráficas de evolución de tráfico generado	19/11/2007	25/11/2007
Migración del protocolo WISPER a Java	26/11/2007	02/12/2007
Desarrollo de las gráficas de evolución de la simulación en rendimiento, paquetes perdidos y retardo medio	03/12/2007	09/12/2007
Desarrollo del módulo de exportación a CSV y XML	10/12/2007	16/12/2007
Desarrollo del módulo de bases de datos (MySQL)	17/12/2007	23/12/2007

Las tareas de liberación y publicidad se detallan en la siguiente tabla:

Tabla 2.3 Tareas de liberación

TAREA	FECHA INICIO	FECHA FIN
Registro del proyecto en SourceForge	23/11/2007	24/11/2007
Preparación del sitio de desarrollo (CVS, Web, listas de correo, etc.)	10/12/2007	16/12/2007
Generar documentación	17/12/2007	23/12/2007
Generar los paquetes binarios y de fuentes	24/12/2007	30/12/2007

Publicación de los paquetes y la documentación	02/01/2008	02/01/2008
Anuncio en la comunidad	03/01/2008	04/12/2008
Mantenimiento (gestión de errores, infraestructura, atender listas de correo, etc.)	Enero 2008	?

A partir de esta planificación inicial en el capítulo de conclusiones se expondrán los objetivos conseguidos y las tareas realizadas.

CAPÍTULO 3. FUNDAMENTOS TEÓRICOS

3.1. Introducción

WISPER es un protocolo MAC para sistemas móviles que utilizan técnicas de acceso CDMA. Se basa en planificar la transmisión de los paquetes que recibe el sistema en función de sus requerimientos BER. El proceso de planificación asigna distintas prioridades a los paquetes en base a las características del tipo de tráfico al cual pertenecen (tiempo de expiración, tasa de transmisión y número de paquetes a transmitir). A continuación esta prioridad se utiliza para calcular el número de paquetes con permiso de transmisión. Finalmente, el procedimiento de asignación se encarga de mapear los paquetes en las ranuras de la trama en función de su prioridad y de sus requerimientos BER.

Mediante este proceso iterativo, se consigue que los paquetes con mayor prioridad se transmitan primero y que ranuras próximas contengan paquetes con requerimientos BER iguales o similares.

3.2. Objetivos

WISPER puede clasificarse como un protocolo ranurado y de asignación bajo demanda. Si tenemos en cuenta que el ancho de banda es un recurso escaso en los sistemas de comunicaciones móviles, los principales objetivos de este protocolo son:

- Maximizar el rendimiento.
- Minimizar la pérdida de paquetes.

A la hora de cumplir dichos objetivos, el protocolo utiliza las siguientes estrategias:

- Para maximizar el rendimiento, WISPER planifica la transmisión de paquetes en función de la clase de tráfico. Concretamente, la ordenación se realiza en base al requerimiento BER más severo y utilizando la máxima tasa de transmisión posible del terminal móvil.
- La pérdida de paquetes se minimiza mediante un esquema de prioridad determinado por el número de paquetes a transmitir y su tiempo de expiración. Las peticiones de transmisión con mayor número de paquetes y cuyo tiempo de vida es menor se transmiten primero. Ambos parámetros vienen determinados por el tipo de tráfico.

De las técnicas de planificación anteriores, se obtiene la primera ventaja del protocolo WISPER; que es poder utilizar el esquema de control de potencia convencional, debido a que los paquetes se asignan en diferentes ranuras en

función de su BER. Por lo tanto, puede utilizarse un único nivel de potencia por ranura, lo cual es bastante más sencillo que usar un esquema con diferentes niveles de potencia al mismo tiempo para cada ranura.

3.3. Escenario

El escenario de uso del protocolo WISPER es una red móvil que soporta varias clases de tráfico multimedia (voz, datos, vídeo, audio, fax, ...). Cada una de estas clases de tráfico presenta unas características y requerimientos distintos, entre los que destacan:

- El esquema de generación de datos, que a grandes rasgos sería constante o variable.
- La duración de la transmisión.
- El tiempo de expiración.
- Los requerimientos necesarios para cumplir una QoS (calidad de servicio) específica:
 - BER o tasa de bits erróneos.
 - PER o tasa de paquetes erróneos.
 - PLR o tasa de paquetes perdidos.

Los terminales móviles generan peticiones de transmisión formadas por lotes o grupos de paquetes. Todos los paquetes pertenecientes a un lote tienen el mismo tiempo de expiración y los terminales móviles operan a su tasa máxima de transmisión.

El ancho de banda disponible se divide en dos bandas, una para el enlace o canal ascendente, y otra para el enlace o canal descendente. En el primero de ellos, los terminales móviles envían peticiones de transmisión a la estación base y en el segundo, es la estación base la que indica como deben transmitir dichos paquetes.

Para cada una de las bandas, el tiempo se divide en tramas de longitud t_f . La longitud de la trama se elige para que coincida con la tasa de llegada de paquetes de la clase de tráfico más utilizada (usualmente voz). La longitud típica de la trama es de 16 ms.

A continuación se pasa a describir un poco más en profundidad cada uno de los enlaces anteriores.

El enlace ascendente

Como muestra la figura 3.1, cada trama se divide en N_p ranuras de datos y en una ranura de petición. En principio, aunque cada ranura de datos puede transportar cualquier tipo de tráfico, el terminal móvil genera simultáneamente información de un solo tipo. Por su parte, la ranura de petición se utiliza para dos

propósitos:

- Alojar peticiones de admisión realizadas por nuevos terminales móviles que quieren acceder a la red.
- Alojar peticiones de transmisión realizadas por terminales actualmente registrados en la red.

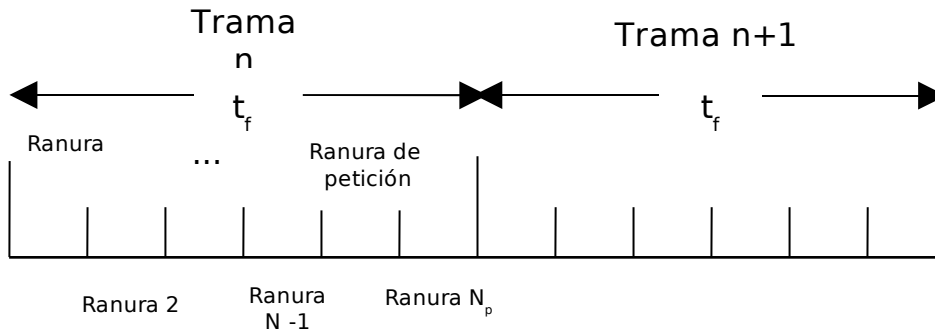


Fig. 3.1 División en tramas y ranuras del enlace ascendente

Cuando un terminal móvil solicita acceso en la red, selecciona aleatoriamente un código PN de una serie de códigos reservados. Mediante este código, envía un paquete de solicitud de admisión en la ranura de petición. La ranura de petición contiene el número de identificación y la máxima tasa de transmisión del terminal móvil, así como información sobre la clase de tráfico que debe ser transmitida y sus máximas tasas de bits y paquetes (BER, PER y PLR).

Pueden ocurrir colisiones en la ranura de petición si se producen dos o más solicitudes de admisión con el mismo código PN, o si el número total de terminales móviles solicitantes simultáneos provoca interferencia mutua. Si no hay colisiones y el paquete se recibe en la estación base, el CAC (controlador de admisión de llamadas) determina si los requerimientos de QoS de los terminales registrados previamente no se degradarán si el nuevo terminal es admitido. En caso afirmativo, el terminal móvil recibe una notificación de aceptación. Esta notificación incluye un número único de identificación y un código PN primario para uso exclusivo del terminal móvil durante el tiempo de conexión.

Después de registrarse en la red, cada vez que el terminal móvil tiene nuevos paquetes preparados para transmitir envía una petición de transmisión a la estación base. Esta petición contiene información sobre el número de paquetes a transmitir y su tiempo de expiración.

El terminal remoto envía la solicitud de transmisión utilizando la ranura de petición o una ranura de datos. El último método puede utilizarse para reducir la contención en la ranura de petición. En ambos casos, las peticiones se transmiten utilizando el código PN primario asignado al terminal móvil. Una vez se ha recibido la petición, la estación base asigna una estructura interna de

datos para realizar su seguimiento. Esta estructura contiene información tal como el identificador del terminal móvil, el tiempo de expiración de los paquetes y el número de paquetes a transmitir. Toda esta información se almacena hasta que los paquetes han sido recibidos satisfactoriamente o hasta que expiran y se descartan.

El enlace descendente

La estación base se encarga del control del tráfico en el canal de bajada. Las tramas del enlace descendente también son de longitud t_f y se dividen en N_p ranuras de datos y una ranura de control. Esta última, cumple las siguientes funciones, complementarias a las que realiza la ranura de petición en el canal de subida:

- Notifica la aceptación a los terminales móviles que previamente solicitaron permiso de admisión.
- Proporciona las instrucciones de transmisión necesarias a los terminales móviles que envían peticiones de transmisión.

La figura 3.2 muestra la estructura en tramas del enlace descendente:

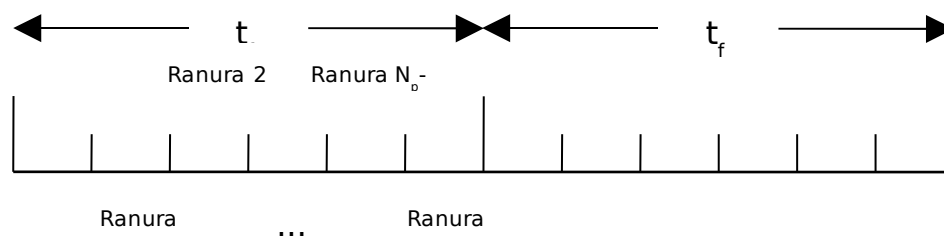


Fig. 3.2 División en tramas y ranuras del enlace descendente

La estación base no necesita enviar el número de identificación cuando intenta comunicarse con un terminal móvil en concreto. Simplemente transmite el mensaje utilizando el mismo código PN que utiliza el terminal remoto. Cuando responde a una petición de transmisión, especifica el número de paquetes que pueden transmitirse en la próxima trama y las ranuras a las cuales están asignados. La ranura de control también puede utilizarse para proporcionar información de acuse de recibo de los paquetes recibidos con éxito en la trama anterior.

La figura 3.3 muestra los tiempos relativos entre las tramas del enlace ascendente y el enlace descendente:

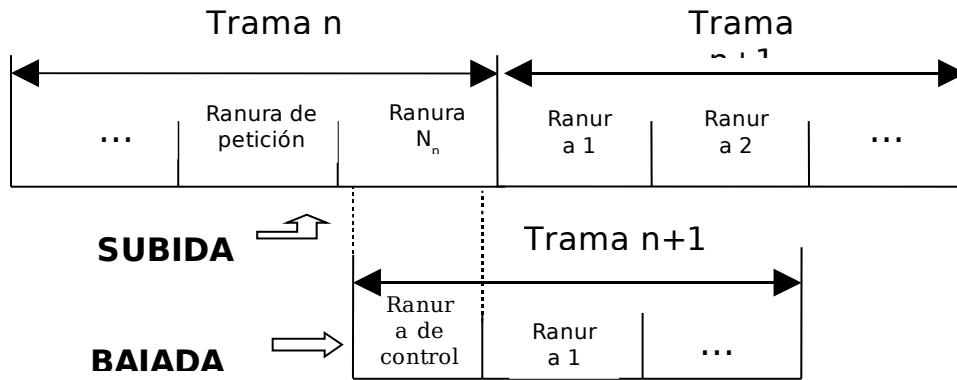


Fig. 3.3 Tiempos relativos entre el enlace ascendente y el descendente

Para el enlace de subida, la ranura de petición no puede ser la última de la trama, ya que la estación base necesita un cierto tiempo para procesar las peticiones y para calcular las asignaciones antes de que empiece la trama de subida. La posición exacta de la ranura de petición dependerá de la capacidad de procesamiento de la estación base.

Por su parte, el canal de bajada comienza con la ranura de control. Debe iniciarse una vez recibida y procesada la ranura de petición de la trama ascendente y debe finalizar antes de que se inicie la siguiente trama para que el terminal móvil tenga tiempo de procesar las respuestas de la estación base.

Se debe tener en cuenta que el tamaño de las ranuras de petición y de control no tiene que ser necesariamente igual al de las ranuras de datos. Normalmente, el tamaño de estas ranuras es menor.

3.4. Consideraciones previas

El protocolo WISPER asigna una tasa BER a las ranuras y planifica la transmisión de paquetes en dichas ranuras con el objetivo de optimizar el ancho de banda. Cada clase de tráfico k_q ($q = 1, 2, \dots$) tiene una especificación de BER máxima $B(k_q)$, que viene determinada por varios factores:

- Ruido térmico.
- Interferencia.
- Potencia de señal recibida.
- Tipo de modulación empleado.
- Esquema FEC utilizado.

Además de los factores anteriores, un elemento de importancia fundamental para determinar el valor del parámetro BER es el número de terminales móviles

que transmiten concurrentemente en el mismo ancho de banda del canal. Esto se debe a que, desde el punto de vista del transmisor CDMA, el resto de transmisiones simultáneas en la misma banda de frecuencia se añade como ruido a la señal propia.

Para usar el ancho de banda disponible de forma eficiente, los paquetes que tienen la misma o similar especificación máxima de BER deben transmitirse en la misma ranura. Por ejemplo, si dos paquetes x e y con el requerimiento de máxima BER tolerable $b_x > b_y$ se asignan para transmitir en la misma ranura, entonces el número máximo de transmisiones simultáneas estará limitado por $S(b_y)$. Agrupando los paquetes con similar especificación BER en la misma ranura se consigue mejorar la eficiencia, ya que los paquetes se agrupan en función de la capacidad restante en la ranura y no únicamente en base al tiempo de expiración o a la prioridad del tipo de tráfico. Es decir, los paquetes con un requerimiento BER menos severo disponen de mayor capacidad en la ranura asignada, ya que no están limitados por aquellos que tienen una BER más restrictiva.

Sin embargo, la asignación de paquetes en ranuras en función de su requerimiento BER no es suficiente para conseguir un rendimiento óptimo. En el momento en el que hay más paquetes en espera que aquellos que pueden ser transmitidos en la próxima trama es necesario desarrollar un mecanismo de prioridades para determinar los que deben transmitirse primero. La selección de estos paquetes puede realizarse de acuerdo a varios criterios:

- Proximidad del tiempo de expiración.
- Prioridad del tipo de tráfico del paquete.
- Requerimientos de QoS, tales como los valores máximos de retardo o tasa de pérdida de paquetes máxima.

En el caso del algoritmo WISPER, se opta por asignar prioridades en función del tiempo de expiración.

En el siguiente apartado, se presenta el desarrollo de un módulo planificador, que permite organizar la transmisión de los paquetes en función de los parámetros anteriores.

3.5. El módulo planificador de paquetes

Es imposible diseñar un módulo planificador óptimo para cada terminal móvil del sistema, debido a la cantidad de factores que intervienen:

- Ancho de banda disponible.
- Número de usuarios.
- Requerimientos concretos de QoS para cada tipo de tráfico.
- Características y limitaciones propias de los terminales físicos.

Normalmente, optimizar alguno de los parámetros anteriores implica empeorar los restantes. Por lo tanto, siempre se debe llegar a un compromiso entre los distintos parámetros que intervienen en el sistema.

En el caso concreto de un sistema de comunicación móvil, el ancho de banda es el recurso más escaso, por lo que el principal objetivo debe ser maximizar la tasa de transmisión y minimizar la pérdida de paquetes. Para cumplir los objetivos anteriores, el módulo planificador debe implementar las siguientes funciones:

- Calcular la prioridad que se asigna a los paquetes.
- Asignar la ranura en que se transmiten dichos paquetes.

El terminal móvil proporciona la información de tiempo de expiración y del número de paquetes a transmitir mediante una petición de transmisión a la estación base. La estación base recibe la petición y la almacena en una cola. Una vez finalizada la trama del enlace ascendente, la estación base utiliza una función de asignación de prioridad para determinar qué paquetes tienen un tiempo de expiración menor y por lo tanto, deben ser a priori transmitidos en la siguiente trama. Este procedimiento no calcula la prioridad de paquetes individuales, sino que obtiene un único valor para todo el conjunto de paquetes que el terminal móvil solicita transmitir. Por lo tanto, en un instante de tiempo determinado, todos los paquetes del grupo tienen valores de prioridad idénticos, aunque esto no significa necesariamente que se envíen en la misma trama.

A continuación, se detalla la implementación del cálculo de prioridades y de la asignación en ranuras.

Cálculo de prioridad

El cálculo de prioridad se realiza de forma dinámica al final de cada trama y se utiliza para limitar el número de paquetes que un terminal móvil puede enviar cuando la célula está muy cargada. Una vez se ha calculado la prioridad de los paquetes, se seleccionan aquellos que tienen un valor más elevado para transmitirlos de forma inmediata.

La prioridad de una petición de transmisión se calcula a partir del número de paquetes a transmitir y de su tiempo de expiración. Mediante estos parámetros, se consigue minimizar la pérdida de paquetes, ya que en situaciones de carga, las peticiones con un tiempo de vida menor y mayor número de paquetes se transmiten primero.

Por lo tanto, para reducir las pérdidas, el valor de prioridad debe cumplir los siguientes requisitos:

- Debe ser inversamente proporcional al número de tramas que restan hasta alcanzar el tiempo de expiración de los paquetes.
- Debe ser directamente proporcional al número de paquetes que forman la petición de transmisión.

El cálculo de prioridad viene determinado por la siguiente expresión:

$$\Phi_b(t) = \begin{cases} \frac{\lceil P_b(t)/M_n \rceil}{F_b(t)}, & \text{si } \frac{\lceil P_b(t)/M_n \rceil}{F_b(t)} \leq N_p \\ N_p, & \text{sino} \end{cases} \quad (3.1)$$

donde:

- $P_b(t)$ representa el número de paquetes que forman la petición de transmisión b en el instante de tiempo t .
- M_n es la tasa máxima de transmisión del terminal móvil n , es decir, el número máximo de paquetes por ranura que puede transmitir simultáneamente.
- $F_b(t)$ representa el número de tramas restantes en el instante de tiempo t hasta la expiración de la petición b .
- N_p indica el número de ranuras en una trama.
- $\Phi_b(t)$ representa la prioridad de la petición b en el instante de tiempo t .

Se observa que la prioridad $\Phi_b(t)$ es directamente proporcional al número de paquetes $P_b(t)$ que forman la petición e inversamente proporcional a la tasa máxima de transmisión M_n y al número de tramas restantes hasta su expiración $F_b(t)$.

Otra forma de evaluar la función anterior, es considerar que el número de paquetes $P_b(t)$ de la petición entre la tasa máxima de transmisión M_n equivalen al mínimo número de ranuras necesarias para transmitir dichos paquetes. Según esta interpretación, la prioridad de la petición es directamente proporcional al número mínimo de ranuras necesarias e inversamente proporcional al número de tramas restantes hasta su expiración. A su vez, el número de ranuras necesarias entre el número de tramas hasta la expiración indica el mínimo número de ranuras que deben asignarse por trama para garantizar que la petición no caduque.

A la hora de aplicar la función de prioridad, el terminal móvil debe cumplir los siguientes requisitos:

- Siempre opera con su tasa máxima de transmisión.
- Transmite en ranuras uniformemente espaciadas en las tramas restantes hasta la expiración de los paquetes.

Para un cierto terminal móvil que cumpla los requerimientos anteriores, la prioridad de la petición de transmisión vendrá en función del número de paquetes que la forman y de su tiempo de vida. Si el número de paquetes a

transmitir es elevado, mucho más que el número de tramas restantes antes de que caduquen, la petición obtendrá una prioridad mayor. Por el contrario, si el tiempo de expiración es elevado, la función asignará una prioridad menor. De la fórmula de prioridad, se observa que el rango de valores está comprendido entre $0 < \Phi_b(t) < N_p$. Si se consume el tiempo de vida de la petición, es decir, cuando $F_b(t) = 0$, la estación base descarta los paquetes y actualiza sus registros.

Después de asignar una prioridad a cada petición, el valor se utiliza para determinar si un terminal móvil debe tener privilegios de transmisión en la próxima trama y si es así, el número de paquetes que puede enviar y su posición. Esta tarea la realiza el procedimiento de acomodación de paquetes.

Procedimiento de acomodación de paquetes

Una vez calculada la prioridad de los paquetes, el procedimiento de acomodación se encarga de seleccionar los paquetes que se enviarán en la próxima trama. Por un lado, calcula el número de paquetes con permiso de transmisión y por otro, el procedimiento de asignación determina la posición exacta dentro de la trama. El objetivo del procedimiento de acomodación es colocar los paquetes con mayor prioridad de tal forma que se maximice el rendimiento en la próxima trama.

Para calcular el número de paquetes con permiso de transmisión, interviene el valor de prioridad de la petición. Este valor contiene información sobre el número máximo de paquetes que el procedimiento de acomodación debería permitir enviar a un terminal móvil bajo condiciones de demanda elevada de ancho de banda. Este número se determina mediante la siguiente expresión:

$$N(b) = \begin{cases} \lfloor \Phi_b(t) \cdot M_n \rfloor, & \text{si } \lfloor \Phi_b(t) \cdot M_n \rfloor < P_b(t) \\ P_b(t), & \text{sino} \end{cases} \quad (3.2)$$

Donde:

- $\Phi_b(t)$ representa la prioridad de la petición b en el instante de tiempo t .
- M_n es la tasa máxima de transmisión del terminal móvil n .
- $P_b(t)$ representa el número de paquetes que forman la petición de transmisión b en el instante de tiempo t .
- $F_b(t)$ representa el número de tramas restantes en el instante de tiempo t hasta la expiración de la petición b .
- $N(b)$ indica el número de paquetes de la petición b que el terminal móvil n puede transmitir en la próxima trama.

La estación base gestiona las peticiones de transmisión mediante una cola de registros de tamaño q . Existe un registro para cada tipo de tráfico k_q presente en

el sistema. La cola de registros está ordenada de forma ascendente a partir del tipo de tráfico con el requerimiento BER más restrictivo. Cada uno de estos registros, se compone a su vez de una cola con las peticiones de transmisión de los terminales móviles. Esta cola está ordenada por prioridad descendente y contiene, además de los valores de prioridad, el número de paquetes a transmitir, el número de paquetes con permiso de transmisión y el tiempo de expiración de la petición.

El procedimiento de acomodación de paquetes recorre los distintos registros en busca de las peticiones que deben transmitirse. Una petición debe transmitirse si contiene paquetes cuya prioridad es mayor o igual al valor de una variable denominada prioridad. Cuando esto ocurre, el acomodador de paquetes intenta asignar los paquetes de la petición en la próxima trama, utilizando para ello el procedimiento de asignación de paquetes.

Al inicio del proceso de acomodación, la variable prioridad toma el máximo valor detectado entre las peticiones que componen los registros. Este valor permanece constante mientras se examinan los distintos registros de datos. Antes de iniciar otra búsqueda, la variable prioridad toma el siguiente máximo valor de entre todas las peticiones que permanecen en el sistema.

Después de que todas las peticiones en los registros hayan sido consideradas para transmitir, se vuelve a calcular su valor de prioridad, considerando sólo los paquetes que no han sido asignados. El procedimiento de acomodación finaliza cuando se han servido todas las peticiones de transmisión o bien cuando no se pueden asignar más paquetes en la próxima trama.

La descripción del pseudocódigo para el procedimiento de acomodación de paquetes se describe en la figura 3.4:

```

PROCEDIMIENTO acomodar_paquetes
INICIO
  PARA w = 0 HASTA w < q HACER
    actualizar_prioridades_en_cola(w);
  FIN
  prioridad = proxima_prioridad(∞);
  MIENTRAS (trama_llena == false) HACER
    PARA x = 0 HASTA x < q HACER
      PARA y = 0 HASTA y < numero_de_peticiones(x) HACER
        prioridad_peticion = obtener_prioridad(y, x);
        SI (prioridad_peticion >= prioridad) ENTONCES
          permitidos = calcular_paquetes(prioridad_peticion);
          asignados = asignar_paquetes(n, Mn, permitidos, kx);
        FIN
      FIN
    FIN
  SI (asignados == 0) ENTONCES return;
  prioridad = proxima_prioridad(prioridad);
  SI (prioridad > 0) ENTONCES continue;
  SINO
    SI (peticiones_por_servir > 0)
      PARA w = 0 HASTA w < q HACER
        actualizar_prioridades_en_cola(w);
      FIN
      prioridad = proxima_prioridad(∞);
      continue;
    FIN
  FIN
  return;
FIN

```

Fig. 3.4 El procedimiento de acomodación de paquetes

Procedimiento de asignación de paquetes

El procedimiento de asignación de paquetes permite determinar si una petición puede servirse en una trama y el número de paquetes que se pueden asignar por ranura. Debido a que es un proceso iterativo para las distintas peticiones pendientes de transmisión, también debe monitorizar en qué ranuras se asignan finalmente los paquetes. El procedimiento requiere cuatro parámetros:

- Número de identificación n del terminal móvil.
- Máxima velocidad de transmisión posible M_n del terminal móvil n .
- Número de paquetes P_n con permiso de transmisión del terminal móvil n .
- Tipo de tráfico k_n que transmite el terminal móvil n .

En función de la clase de tráfico de los paquetes que deben ser acomodados y de las ranuras disponibles en la próxima trama, el proceso intenta asignar los paquetes de acuerdo con los siguientes criterios y en el siguiente orden:

- Asignación en ranuras vacías o en ranuras que contienen paquetes con el mismo tipo de tráfico.
- Asignación en ranuras que contienen paquetes con requerimientos de BER más severos.
- Asignación en ranuras que contienen paquetes con requerimientos de BER menos severos.

Cada uno de los criterios anteriores corresponde a una fase del procedimiento de asignación de paquetes.

En la primera fase, se buscan ranuras vacías en el trama o bien ranuras que contengan el mismo tipo de tráfico k_n . Si hay ranuras con tales características, el proceso de asignación intenta colocar todos los paquetes P_n en dichas ranuras. Si esto no es posible, el procedimiento busca otras ranuras vacías u otras ranuras con el mismo tipo de tráfico para acomodar los paquetes restantes. Si se llega a la última ranura y hay todavía paquetes por asignar, se procede con la segunda fase del algoritmo.

Para la segunda fase, el procedimiento de asignación busca una ranura que contenga paquetes de la clase de tráfico k_{n-1} , donde el tipo de tráfico k_{n-1} tiene un requerimiento de BER más severo que el tráfico de tipo k_n . Si se encuentra dicha ranura, el algoritmo intenta acomodar en ella todos los paquetes restantes. Si todavía quedan paquetes por asignar y todas las ranuras están llenas o tienen paquetes de otros tipos de tráfico, el procedimiento busca ranuras que tengan paquetes con tipos de tráfico k_{n-2}, \dots, k_1 . Esta fase finaliza cuando se han asignado todos los paquetes o bien cuando las ranuras con tipo de tráfico k_1 han sido consideradas.

Si todavía quedan paquetes por asignar, el procedimiento continua con la tercera fase. El algoritmo recorre la trama buscando ranuras con tráfico de tipo k_{n+1} , es decir, con un requerimiento BER menos severo. Si se encuentra dicha ranura, el procedimiento de asignación verifica que la ranura puede convertirse al tipo de tráfico k_n . Para esto, se verifica que no se excede el número máximo de paquetes permitidos para soportar el tráfico de clase k_n . Lógicamente, la ranura se convierte a tipo de tráfico k_n sólo si hay espacio para acomodar por lo menos un paquete más. De forma similar a la fase anterior, si se alcanza la última ranura y aún hay paquetes por asignar, el procedimiento busca ranuras de tipos de tráfico $k_{n+1}, k_{n+2}, \dots, k_q$, y verifica si pueden convertirse a tipo de tráfico k_n para asignar los paquetes restantes.

El procedimiento de asignación termina cuando todos los paquetes de una petición han sido asignados o bien cuando se llega al final de la tercera fase.

La descripción del pseudocódigo para el procedimiento de asignación de paquetes se presenta en la figura 3.5:

```

FUNCION asignar_paquetes(n, Mn, Pn, kx)
INICIO
  PARA i = 0 HASTA i < slots_en_trama HACER
    SI (slot_lleno(x) == false) ENTONCES
      SI (slot_vacio(x) == true) ENTONCES
        asignar_trafico(x, kn);
        asignar_capacidad(x, kn)
      FIN
      SI (obtener_trafico(x) == kn) ENTONCES
        asignados = calcular_paquetes_asignados(x, Pn, Mn);
        totales = actualizar_asignados_totales(x, asignados);
        Pn = actualizar_paquetes_restantes(asignados);
        actualizar_tamaño_slot(x, asignados);
        SI (Pn == 0) ENTONCES
          return totales;
        FIN
      FIN
    FIN
  FIN

  PARA i = 0 HASTA i < slots_en_trama HACER
    SI (slot_lleno(x) == false) ENTONCES
      SI (obtener_capacidad(x) < BER(kx)) ENTONCES
        asignados = calcular_paquetes_asignados(x, Pn, Mn);
        totales = actualizar_asignados_totales(x, asignados);
        Pn = actualizar_paquetes_restantes(asignados);
        actualizar_tamaño_slot(x, asignados);
        SI (Pn == 0) ENTONCES
          return totales;
        FIN
      FIN
    FIN
  FIN

  PARA i = 0 HASTA i < slots_en_trama HACER
    SI (slot_lleno(x) == false) ENTONCES
      SI (obtener_capacidad(x) > BER(kx)) ENTONCES
        SI (convertir_tipo_slot(x, kx) == true)
          asignados = calcular_paquetes_asignados(x, Pn, Mn);
          totales = actualizar_asignados_totales(x, asignados);
          Pn = actualizar_paquetes_restantes(asignados);
          actualizar_tamaño_slot(x, asignados);
        FIN
        SI (Pn == 0) ENTONCES
          return totales;
        FIN
      FIN
    FIN
  FIN
  FIN
  return totales;

```

Fig. 3.5 El procedimiento de asignación de paquetes

3.6. Ejemplo

A continuación se presenta un ejemplo práctico donde se muestra el funcionamiento del algoritmo WISPER. Se describen paso a paso todos los procedimientos realizados por la estación base para conseguir asignar los paquetes en las diferentes ranuras de la trama de una manera eficiente.

Se parte de la situación inicial en la que la estación base recibe las peticiones de transmisión de los diferentes terminales móviles. En este caso concreto, llegan 7 peticiones, con un total de 32 paquetes que solicitan permiso de transmisión. Estos paquetes pertenecen a 4 tipos de tráfico diferentes, denominados A, B, C y D. Los paquetes que pertenecen al tipo de tráfico A poseen el requerimiento BER más severo, lo cual se traduce en que sólo hasta $S = 4$ paquetes pueden ser transmitidos en la misma ranura, si quiere respetarse la tasa de bits erróneos de dicho tráfico. Para las clases de tráfico B, C y D, el máximo número de paquetes por ranura está limitado a $S = 5$, $S = 6$ y $S = 7$, respectivamente. Otro parámetro importante en el sistema es la tasa de transmisión máxima, denominada M. Los valores de la tasa de transmisión para A, B, C y D son $M = 3$, $M = 4$, $M = 4$ y $M = 7$, respectivamente. Suponemos que todos los terminales móviles son capaces de transmitir a tasa máxima. Para el caso del ejemplo, las tramas ascendente y descendente constan ambas de 6 ranuras.

Una vez la estación base ha recibido todas las peticiones de transmisión, crea un registro para cada tipo de tráfico recibido. Dentro de cada uno de estos registros se almacenan las peticiones de transmisión de los distintos terminales móviles que transmiten el mismo tipo de tráfico. Los registros se ordenan de forma ascendente en función de su BER. A continuación, la estación base calcula la prioridad de cada petición y el número de paquetes con permiso de transmisión. Finalmente, vuelve a ordenar las peticiones de cada registro de forma descendente en función del valor de prioridad obtenido. Los valores de prioridad y paquetes con permiso de transmisión se obtienen de las expresiones 3.1 y 3.2. Cabe destacar que para que la estación base realice estos cálculos iniciales necesita recibir del terminal móvil su identificador, el tipo de tráfico que solicita acceso, el número de paquetes a transmitir y su tiempo de expiración. La tabla 3.1 muestra todos estos parámetros, una vez calculadas las prioridades y los paquetes con permiso de transmisión.

Al principio del procedimiento de asignación de paquetes, se determina el valor de prioridad más elevado entre todos los registros. Una vez se han considerado todas las peticiones con mayor prioridad, se obtiene el siguiente valor de prioridad y se evalúan las peticiones asociadas. Este proceso se repite hasta que se evalúan todas las prioridades. En este punto, vuelven a calcularse las prioridades de las peticiones restantes y el número de paquetes con permiso de transmisión en función de los paquetes que quedan por asignar. Este procedimiento finaliza si el número de paquetes asignados en la evaluación de prioridades previa es igual a cero o cuando se han asignado todas las peticiones de transmisión. En cualquier caso, el procedimiento también finaliza si se llena la trama y por lo tanto, no pueden asignarse más paquetes a las ranuras.

Tabla 3.1 Parámetros del ejemplo

Tipo de tráfico: A Tasa de transmisión M = 3 (paquetes/slot) Máximo número de paquetes por "slot" S = 4 (paquetes/slot)					
Identificador móvil: A-1		Identificador móvil: A-2			
TO = 1	P = 4	TO = 2	P = 5		
Prioridad	Paquetes	Prioridad	Paquetes		
2	4	1	3		
-	-	0'5	2		
Tipo de tráfico: B Tasa de transmisión M = 4 (paquetes/slot) Máximo número de paquetes por "slot" S = 5 (paquetes/slot)					
Identificador móvil: B-1					
TO = 1		P = 3			
Prioridad		Paquetes			
1		3			
-		-			
Tipo de tráfico: C Tasa de transmisión M = 4 (paquetes/slot) Máximo número de paquetes por "slot" S = 6 (paquetes/slot)					
Identificador móvil: C-1		Identificador móvil: C-2		Identificador móvil: C-3	
TO = 2	P = 5	TO = 2	P = 2	TO = 3	P = 6
Prioridad	Paquetes	Prioridad	Paquetes	Prioridad	Paquetes
1	4	0'5	2	0'66	4
0'5	1	-	-	0'33	2
Tipo de tráfico: D Tasa de transmisión M = 7 (paquetes/slot) Máximo número de paquetes por "slot" S = 7 (paquetes/slot)					
Identificador móvil: D-1					
TO = 1			P = 7		
Prioridad			Paquetes		
1			5		
-			-		

Para el primer paso, que corresponde a la tabla 3.2, sólo se consideran las

peticiones de transmisión que tienen una prioridad igual a 2. Únicamente la petición A-1 tiene esta prioridad. Los 4 paquetes con permiso de transmisión se asignan a las ranuras 1 y 2, estableciendo que el máximo número de paquetes que pueden transmitirse en dichas ranuras son $S = 4$. Al finalizar este paso, todos los paquetes de la petición A-1 se han asignado a la trama.

Tabla 3.2 Primer paso del procedimiento de acomodación

Prioridad ≥ 2					
Cálculos de prioridad = 1					
A-1					
A-1					
A-1	A-1				
Capacidad	Capacidad	Capacidad	Capacidad	Capacidad	Capacidad
4	4	N/A	N/A	N/A	N/A

En el segundo paso, que corresponde a la tabla 3.3, el siguiente valor de prioridad a evaluar es 1. La petición A-2 tiene esa prioridad. Por lo tanto, se buscan ranuras vacías o ranuras con el mismo tipo de tráfico para asignar los paquetes con permiso de transmisión, que en este caso son 3. Estos paquetes se asignan a las ranuras 1 y 2, la primera de los cuales pasa a estar llena. A continuación se busca otra petición con prioridad igual a 1. La siguiente es la petición B-1, que tiene permiso para transmitir todos los paquetes, en este caso 3. La siguiente ranura vacía es la tercera, por lo que todos los paquetes de la petición B-1 se asignan a ella. La capacidad de la tercera ranura queda limitada a $S = 5$. La siguiente petición con prioridad igual a 1 es C-1. Esta petición está formada por 5 paquetes, aunque inicialmente sólo 4 tienen permiso de transmisión y se asignan a la cuarta ranura, que también estaba vacía. Tras esta asignación, el máximo número de paquetes que pueden transmitirse en dicha ranura es $S = 6$. Finalmente, la última petición con prioridad igual a 1 es D-1, que tiene 7 paquetes con permiso de transmisión. Todos ellos se asignan a la siguiente ranura vacía, en este caso la quinta, que pasa a estar llena.

Tabla 3.3 Segundo paso del procedimiento de acomodación

Prioridad ≥ 1					
Cálculos de prioridad = 1					
				D-1	
				D-1	
				D-1	
A-2			C-1	D-1	
A-1	A-2	B-1	C-1	D-1	
A-1	A-2	B-1	C-1	D-1	
A-1	A-1	B-1	C-1	D-1	
Capacidad	Capacidad	Capacidad	Capacidad	Capacidad	Capacidad
4	4	5	6	7	N/A

El tercer paso corresponde a la tabla 3.4 y se evalúan las peticiones con prioridad 0'66. Únicamente la petición C-3 tiene dicha prioridad, por lo que los 4 paquetes con permiso de transmisión se asignan a la última ranura vacía, la sexta. El máximo número de paquetes que pueden transmitirse en dicha ranura pasa a estar limitado a $S = 6$.

Tabla 3.4 Tercer paso del procedimiento de acomodación

Prioridad $\geq 0'66$					
Cálculos de prioridad = 1					
				D-1	
			C-3	D-1	
			C-3	D-1	
A-2			C-1	D-1	
A-1	A-2	B-1	C-1	D-1	
A-1	A-2	B-1	C-1	D-1	C-3
A-1	A-1	B-1	C-1	D-1	C-3
Capacidad	Capacidad	Capacidad	Capacidad	Capacidad	Capacidad
4	4	5	6	7	6

En el cuarto paso, que corresponde a la tabla 3.5, se evalúa el valor de prioridad 0'5. La única petición con dicho valor de prioridad es C-2. Como ya no quedan ranuras vacías, el procedimiento de asignación busca ranuras con el mismo tipo

de tráfico que no estén llenas. La sexta ranura tiene estas características y los 2 paquetes con permiso de transmisión de C-2 (que son todos los que la forman) se asignan en ella.

Tabla 3.5 Cuarto paso del procedimiento de acomodación

Prioridad $\geq 0'5$ Cálculos de prioridad = 1					
				D-1	
			C-3	D-1	
			C-3	D-1	
A-2			C-1	D-1	C-2
A-1	A-2	B-1	C-1	D-1	C-2
A-1	A-2	B-1	C-1	D-1	C-3
A-1	A-1	B-1	C-1	D-1	C-3
Capacidad	Capacidad	Capacidad	Capacidad	Capacidad	Capacidad
4	4	5	6	7	6

Llegados a este punto, se han evaluado todas las prioridades, pero aún quedan paquetes por asignar y espacio en la trama. Por lo tanto, vuelven a calcularse las prioridades y el número de paquetes con permiso de transmisión en función de los paquetes restantes.

En el quinto paso, que corresponde a la tabla 3.6, se evalúa la prioridad 0'5. La petición A-2 tiene esta prioridad y 2 paquetes por asignar. En primer lugar, se buscan ranuras con el mismo tipo de tráfico que no estén llenas. La segunda ranura tiene estas características, aunque sólo cabe un paquete más. El siguiente paso es buscar ranuras con requerimientos BER más severos. Sin embargo, el tipo de tráfico A es el que tiene el requerimiento BER más severo de todos los presentes en el sistema. Por lo tanto, se pasa a buscar ranuras con un requerimiento BER menos severo y que puedan convertirse al tipo de tráfico A. La tercera ranura cumple estas características y en ella se asigna el paquete restante. Finalmente, la petición C-1 también tiene prioridad 0'5 y 1 paquete por asignar. Como la sexta ranura contiene paquetes del mismo tipo de tráfico, el paquete restante de la petición C-1 se asigna en ella.

Tabla 3.6 Quinto paso del procedimiento de acomodación

Prioridad $\geq 0'5$ Cálculos de prioridad = 2					
				D-1	
			C-3	D-1	
			C-3	D-1	C-1
A-2	A-2	A-2	C-1	D-1	C-2
A-1	A-2	B-1	C-1	D-1	C-2
A-1	A-2	B-1	C-1	D-1	C-3
A-1	A-1	B-1	C-1	D-1	C-3
Capacidad	Capacidad	Capacidad	Capacidad	Capacidad	Capacidad
4	4	4	6	7	6

Finalmente, el sexto y último paso corresponde a la tabla 3.7 y se evalúa el valor de prioridad 0'33. La única petición con dicha prioridad es C-3 y le restan 2 paquetes por asignar. Sin embargo, únicamente es posible asignar 1 paquete en la sexta ranura. Tras esta última asignación, la trama pasa a estar llena y ya no es posible asignar nuevos paquetes. El paquete restante de la petición C-3 tiene un tiempo de expiración de 3 tramas, por lo que estará disponible para ser asignado en la siguiente trama.

Tabla 3.7 Sexto paso del procedimiento de acomodación

Prioridad $\geq 0'33$ Cálculos de prioridad = 2					
				D-1	
			C-3	D-1	C-3
			C-3	D-1	C-1
A-2	A-2	A-2	C-1	D-1	C-2
A-1	A-2	B-1	C-1	D-1	C-2
A-1	A-2	B-1	C-1	D-1	C-3
A-1	A-1	B-1	C-1	D-1	C-3
Capacidad	Capacidad	Capacidad	Capacidad	Capacidad	Capacidad
4	4	4	6	7	6

A partir de este ejemplo podemos comprobar la eficiencia del protocolo WISPER.

En este ejemplo concreto, se han asignado 31 paquetes de un total de 32. Una asignación aleatoria de los paquetes en función de su prioridad podría haber acabado en el peor caso en una asignación de 24 paquetes, con lo que la ganancia obtenida mediante WISPER está entomo al 23%.

Sin embargo, debe tenerse en cuenta la complejidad en el proceso de asignación de paquetes. Esta complejidad repercute en la potencia de cálculo que necesita la estación base, ya que todos los procedimientos anteriores deben realizarse en el decalado existente entre la trama ascendente y la trama descendente, que es del orden de unos pocos milisegundos.

3.7 Tipos de tráfico

A la hora de evaluar el rendimiento del protocolo, se han utilizado diferentes tipos de tráfico. Se han escogido aquellos que en principio se supone tendrán más peso en las redes móviles multimedia. Cada uno de estos modelos tiene unas características y requerimientos concretos, en función del tipo de información que generan y de la tasa de bit característica de la fuente. Los tipos de tráfico son los siguientes:

- Voz
- Vídeo VBR
- Vídeo CBR
- Audio CBR
- Datos ABR
- Correo ABR

A continuación, se pasa a describir cada uno de los tipos de tráfico anteriores con mayor profundidad.

Voz

Este modelo simula los patrones de habla generados durante una conversación. El comportamiento de la fuente se aproxima por una sucesión de intervalos de habla y de silencio. Estos intervalos son producto de los momentos de habla, pausa y escucha que se producen en la comunicación vocal. Es por lo tanto, un modelo de tipo ráfaga (“burst”).

Una primera aproximación da lugar a un modelo de dos estados, denominado detector de habla lento. Estos dos estados se corresponden con los intervalos de habla y de pausa de la fuente, donde el cambio de estado viene determinado por una cierta probabilidad de cada uno.

Sin embargo, puede obtenerse un modelo más exacto teniendo en cuenta que los intervalos de habla se dividen a su vez en intervalos de corta actividad de

habla y de silencio (espúreos). Estos pequeños intervalos de silencio hacen que el período de habla no sea totalmente continuo. Esta nueva aproximación se denomina detector de habla rápido y puede modelarse mediante un sistema de Markov de tres estados. La figura 3.6 muestra ambos modelos:

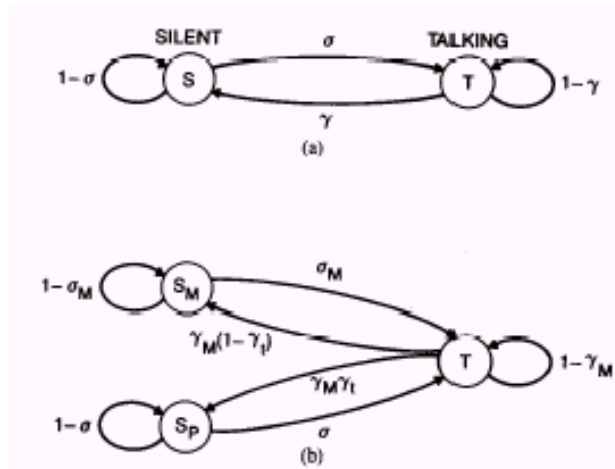


Fig. 3.6 Modelo de estados del detector de habla lento (a) y rápido (b)

Los intervalos de habla, de silencio y sus respectivos espúreos presentan duraciones distribuidas de forma exponencial y son estadísticamente independientes unos de los otros. Para modelar una distribución exponencial podemos utilizar la siguiente expresión:

$$x = -\beta \cdot \ln R \tag{3.3}$$

Donde:

- β es la media de la distribución exponencial.
- R es un número aleatorio uniforme entre 0 y 1.

Las duraciones medias de los intervalos de habla y silencio, tanto principales como espúreos se muestra en la tabla 3.8.

Tabla 3.8 Duraciones medias de los intervalos de habla y silencio, tanto principales como espúreos

		Duración media (s)
Intervalo principal	Habla	$t_1 = 1$
	Silencio	$t_2 = 1'35$

Intervalo espúreo	Habla	$t_3 = 0'275$
	Silencio	$t_4 = 0'050$

La duración media del intervalo principal de habla es t_1 y la duración del intervalo principal de silencio es t_2 . Las duraciones medias de los intervalos espúreos de habla y de silencio son t_3 y t_4 , respectivamente. Los valores numéricos se obtuvieron mediante mediciones experimentales. El cambio de estado se produce al finalizar cada ranura de duración τ . En ese momento, el terminal móvil examina la salida del detector de habla en busca del estado. En caso de que el estado sea de habla, el terminal utiliza un codificador de voz que genera una tasa de bit de 16'5 Kbps. Por lo tanto, el terminal móvil genera una petición de transmisión por trama con un cierto número de paquetes, que está en función del número de estados de habla que se hayan producido. Estas peticiones se almacenan en la cola de entrada y se van actualizando hasta que finaliza la duración media de la comunicación.

Como se observa en la figura 3.6, en el detector de habla rápido intervienen tres estados:

- Habla (T)
- Silencio principal (S_p)
- Mini silencio (S_m)

La probabilidad de transición de S_p a T durante cualquier ranura se obtiene de la expresión 3.4:

$$\sigma = 1 - e^{-\tau/t_2} \quad (3.4)$$

Los intervalos espúreos de silencio tienen una duración media de t_4 y la probabilidad de transición desde el estado S_m a T es:

$$\sigma_M = 1 - e^{-\tau/t_4} \quad (3.5)$$

La duración media de los mini intervalos de habla es t_3 y la probabilidad de finalice en cualquier ranura es:

$$\gamma_M = 1 - e^{-\tau/t_3} \quad (3.6)$$

El número medio de intervalos espúreos de habla en cada intervalo principal es la relación $t_1/(t_3+t_4)$. Por lo tanto, la probabilidad de que cualquier intervalo

espúreo de habla sea el último dentro del intervalo principal es:

$$\gamma_t = \frac{t_3 + t_4}{t_1} \quad (3.7)$$

A partir de las expresiones anteriores, la probabilidad de transición de T a S_m es $\gamma_m \cdot (1 - \gamma_t)$ y la probabilidad de transición de T a S_p es $\gamma_m \cdot \gamma_t$.

El detector de dos estados puede considerarse como un caso particular del detector de tres estados. Si no se detectan los intervalos espúreos de silencio, es decir $t_4 = 0$, y no se distingue entre períodos de habla principales y espúreos ($t_1 = t_3$), entonces $\gamma_t = 1$ y las dos figuras son equivalentes.

Finalmente, en la tabla 3.9 se muestran el resto de parámetros y su valor numérico.

Tabla 3.9 Valores numéricos para el tipo de tráfico voz

Parámetro	Valor
Tasa de bit	16'5 Kbps
Duración media de la transmisión	180 s
BER máxima	10^{-3}
Máximo número de paquetes por ranura	15
Tiempo de expiración de los paquetes	2 tramas
Tasa de transmisión máxima	1 paquete/ranura

Vídeo VBR (Variable Bit Rate)

Este modelo simula las tasas de bit propias de las señales de videoteléfono y videoconferencia. En este caso, la fuente produce múltiples estados, donde cada estado genera un flujo de bits continuo durante un cierto período de tiempo. Los valores de tasa de bit para los diferentes estados se obtienen de una distribución exponencial (expresión 3.3). Esta distribución define un valor medio, un valor máximo y un valor mínimo de tasa de bit. La duración de cada uno de los distintos estados es estadísticamente independiente y exponencialmente distribuida, con una duración media de 160 ms. La duración total de la transmisión también presenta una distribución exponencial, con una media de 180 s.

Desde el punto de vista del simulador, las peticiones de transmisión se almacenan en la cola de entrada con una cierta tasa de transmisión, que permanece invariable mientras dura el estado. Una vez ha finalizado, se pasa

aleatoriamente a un nuevo estado, que define una nueva tasa de bit. La petición permanece en la cola de entrada hasta que expira el tiempo de transmisión. Durante el proceso anterior, la cola de registros captura las peticiones de transmisión de la cola de entrada para procesarlas. La tabla 3.10 resume los valores numéricos usados para el modelo:

Tabla 3.10 Valores numéricos para el tipo de tráfico vídeo VBR

Parámetro	Valor
Duración media de la transmisión	180 s
Duración media del estado	160 ms
Tasa de bit mínima	120 Kbps
Tasa de bit máxima	420 Kbps
Tasa de bit media	239 Kbps
BER máxima	10^{-6}
Máximo número de paquetes por ranura	6
Tiempo de expiración de los paquetes	3 tramas
Tasa de transmisión máxima	5 paquetes/ranura

Vídeo CBR (Constant Bit Rate)

En este modelo se produce un flujo constante de bits. El tiempo de transmisión está distribuido exponencialmente, con un media igual a 180 s. La tasa de transmisión media es igual a 220 Kbps. De cara al simulador, una petición de transmisión de este tipo de tráfico genera un número de paquetes constante a lo largo de las tramas y su duración se obtiene mediante una distribución exponencial en función del valor medio. Los principales parámetros se muestran en la tabla 3.11.

Tabla 3.11 Valores numéricos para el tipo de tráfico vídeo CBR

Parámetro	Valor
Tasa de bit	220 Kbps
Duración media de la transmisión	180 s
BER máxima	10^{-5}
Máximo número de paquetes por ranura	8
Tiempo de expiración de los paquetes	3 tramas

Tasa de transmisión máxima	4 paquetes/ranura
----------------------------	-------------------

Audio CBR (Constant Bit Rate)

Este modelo también se caracteriza por generar un flujo constante de bits, en este caso de audio digital FM. Los parámetros son una tasa de bit igual a 128 Kbps y una duración media de 360 s obtenida mediante una distribución exponencial. Por lo tanto, de cara al simulador, este tipo de tráfico también genera un número constante de paquetes por trama durante una duración exponencialmente distribuida en función de su valor medio. Las principales características se resumen en la tabla 3.12:

Tabla 3.12 Valores numéricos para el tipo de tráfico audio CBR

Parámetro	Valor
Tasa de bit	128 Kbps
Duración de la transmisión	360 s
BER máxima	10^{-4}
Máximo número de paquetes por ranura	10
Tiempo de expiración de los paquetes	6 tramas
Tasa de transmisión máxima	6 paquetes/ranura

Datos ABR (Available Bit Rate)

Se trata de tráfico de tipo ABR ("Available Bit Rate") en el que las peticiones de transmisión se procesan en función de los recursos disponibles en la red, ya que el tiempo de expiración es poco significativo. La estación base captura las peticiones en la cola de entrada y las envía a la cola interna, borrándolas de la primera cola una vez ha finalizado la trama. La longitud de los datos de un mensaje se distribuye exponencialmente, con una media de 30 Kbytes. El tiempo de expiración viene determinado por un factor de ponderación y por la cantidad de paquetes que se envían en la petición de transmisión. Los principales parámetros se muestran en la tabla 3.13.

Tabla 3.13 Valores numéricos para el tipo de tráfico datos ABR

Parámetro	Valor
Tamaño del mensaje	30 kbytes

BER máxima	10^{-9}
Máximo número de paquetes por ranura	4
Tiempo de expiración de los paquetes	$\lceil 2 * \text{paquetes_totales} \rceil$ tramas
Tasa de transmisión máxima	4 paquetes/ranura

Correo ABR (Available Bit Rate)

Este modelo de tráfico es idéntico al tráfico de datos ABR. Como en el tipo de tráfico anterior, la estación base recibe peticiones de transmisión en las que se indica el tamaño del mensaje. La estación base almacena dichas peticiones en la cola interna, donde se procesan en función de los recursos disponibles. Una vez en la cola interna, las peticiones se eliminan de la cola de entrada. El tamaño medio de los mensajes está entorno a los 2900 bytes y también presenta una distribución exponencial. El tiempo de expiración de los paquetes viene determinado por un factor de ponderación y por la cantidad de paquetes que contiene el mensaje. Los principales parámetros se muestran en la tabla 3.14.

Tabla 3.14 Valores numéricos para el tipo de tráfico correo ABR

Parámetro	Valor
Tamaño del mensaje	2900 bytes
BER máxima	10^{-9}
Máximo número de paquetes por ranura	4
Tiempo de expiración de los paquetes	$\lceil 50 * \text{paquetes_totales} \rceil$ tramas
Tasa de transmisión máxima	1 paquete/ranura

CAPÍTULO 4. DISEÑO E IMPLEMENTACIÓN

4.1. Diseño

A la hora de escoger el lenguaje de programación para implementar el simulador MACsim, el principal factor a tener en cuenta ha sido la portabilidad del mismo. El objetivo era conseguir una aplicación que pudiese ejecutarse en diferentes entornos (principalmente sistemas GNU/Linux y Windows) con cambios mínimos. El hecho de disponer de un software multiplataforma aumenta sus posibilidades de difusión.

Finalmente se ha optado por Java. Este lenguaje utiliza el paradigma de programación Orientada a Objetos (POO). La POO es una metodología de programación muy adecuada para diseños complejos, en los cuales intervienen múltiples elementos, como es nuestro caso: colas de datos, tramas, ranuras, diferentes tipos de tráfico, etc. Otras ventajas son que proporciona una amplia API de clases que simplifica el desarrollo y aporta el sistema de documentación JavaDoc que permite documentar el código fuente mediante comentarios. La principal desventaja es que Java es un lenguaje interpretado y por lo tanto, es mucho menos eficiente que los lenguajes compilados, como por ejemplo C++.

Existen multitud de entornos de desarrollo y compiladores para el lenguaje Java. Entre los entornos de desarrollo, dos de los más extendidos son Eclipse (Eclipse Foundation) y NetBeans (SUN Microsystems). Se ha optado por el primero de ellos, ya que es un IDE de licencia libre y quizás está más extendido que el segundo. Esto ofrece la ventaja de que es más fácil involucrar a desarrolladores si éstos conocen el entorno de desarrollo. Otras ventajas de Eclipse es que proporciona un editor visual para crear interfaces gráficas de usuario, conectores XML, de bases de datos, integra la gestión del código mediante un cliente CVS y permite realizar pruebas unitarias con JUnit. Respecto a las herramientas de compilación, depuración, documentación e intérprete se ha utilizado el JDK 1.6 de SUN.

En la tabla 4.1 se adjuntan las principales características del simulador MACsim.

Tabla 4.1 Características hardware y software del simulador MACsim

Lenguaje de programación	Java
Compilador	JDK 1.6
Entorno de desarrollo	Eclipse Europa
Plataforma software	Debian Etch
Plataforma hardware	Intel Pentium 1.73 GHz 1GB RAM

4.2. Implementación

Se han definido los siguientes paquetes para agrupar las diferentes clases implementadas y facilitar la reutilización futura del código:

- **maccsim**

Es el núcleo del simulador. Contiene las clases para modelar todos los objetos que manipula el protocolo (tramas, peticiones de transmisión, ranuras, colas de entrada, etc.), así como las funcionalidades que implementa. Además, se han desarrollado varios tipos de tráfico para poder modelar desde entornos que corresponderían al actual GSM, donde la voz y los SMS son los servicios predominantes, hasta los futuros entornos multimedia de UMTS, donde la transmisión de vídeo digital y datos tendrán un papel esencial. También incluye algunas clases auxiliares para tener facilidades de traza, estadísticas, generación de números aleatorios, mensajes (de cara a simplificar la traducción), etc.

- **maccsimgui**

Contiene todas las clases que implementan el interfaz gráfico de la aplicación, los diferentes diálogos, menús, etc.

- **xygraph**

Contiene las clases que permiten dibujar las gráficas. Esto son los ejes, títulos, etiquetas, marcas, datos a dibujar, etc.

En las siguientes tablas se enumeran las clases de cada uno de los paquetes anteriores que componen el simulador MACsim.

Tabla 4.2 Clases y descripción del paquete maccsim.

CLASE	DESCRIPCIÓN
About.java	Define el nombre de la aplicación, versión, copyright, website del proyecto, etc.
ABRTraffic.java	Tipo de tráfico ABR (Available Bit Rate).
ABRTxRequest.java	Petición de transmisión del tipo de tráfico ABR.
Batch.java	Grupo de paquetes pertenecientes al mismo terminal móvil.
CBRTraffic.java	Tipo de tráfico CBR (Constant Bit Rate).
CBRTxRequest.java	Petición de transmisión del tipo de tráfico CBR.

Frame.java	Cada una de las tramas en las que se divide el enlace ascendente
Level.java	Enumeración con los diferentes tipos de mensajes que proporciona la utilidad de traza.
Logger.java	Utilidad para trazar los pasos del simulador.
Messages.java	Define los diferentes mensajes que aparecen en la aplicación.
Msg.java	Enumeración con los diferentes códigos de los mensajes.
PacketSlot.java	Cada uno de los paquetes que se almacenan en una ranura.
RandNumGen.java	Generador de números aleatorios.
Record.java	Registro almacenado en la cola interna.
RecordQueue.java	Cola interna de registros.
SimState.java	Enumeración con los diferentes estados de la simulación (inicio, en ejecución o parada).
Slot.java	Subdivisión de las tramas en las cuales se almacenan los paquetes (ranuras)
Statistics.java	Utilidad para generar estadísticas.
Traffic.java	Tipo de tráfico genérico.
TrafficManager.java	Gestor y generador de todos los tipos de tráfico presentes en el sistema.
TxRequest.java	Petición de transmisión realizada por un terminal móvil.
TxRequestQueue.java	Cola de entrada donde se almacenan las peticiones de los terminales móviles.
VBRTraffic.java	Tipo de tráfico VBR (Variable Bit Rate).
VBRTxRequest.java	Petición de transmisión del tipo de tráfico VBR.
VoiceState.java	Enumeración con los diferentes estados que se producen durante el proceso de hablar.
VoiceTraffic.java	Tipo de tráfico voz.
VoiceTxRequest.java	Petición de transmisión del tipo voz.
WisperEngine.java	Motor del protocolo WISPER.

Tabla 4.3 Clases y descripción del paquete macsimgui.

CLASE	DESCRIPCIÓN
AboutDialog.java	Diálogo con información de la aplicación.
ABRTrafficDialog.java	Diálogo para configurar el tipo de tráfico ABR.
CBRTrafficDialog.java	Diálogo para configurar el tipo de tráfico CBR.
FrameDialog.java	Diálogo para configurar las tramas.
LicenseDialog.java	Diálogo con información de la licencia.
LoggerDialog.java	Diálogo para configurar la utilidad de traza.
MACsimGUI.java	Clase principal de la interfaz gráfica.
NumericField.java	Permite introducir campos numéricos en los diálogos.
SlotDialog.java	Diálogo para configurar las ranuras.
StatisticsDialog.java	Diálogo para configurar la utilidad de estadísticas.
TrafficManagerDialog.java	Diálogo para configurar el gestor/generador de tráfico.
VBRTrafficDialog.java	Diálogo para configurar el tipo de tráfico VBR.
VoiceTrafficDialog.java	Diálogo para configurar el tipo de tráfico voz.

Tabla 4.4 Clases y descripción del paquete xygraph.

CLASE	DESCRIPCIÓN
Axis.java	Eje genérico.
Dataltem.java	Cada uno de los elementos de datos que se muestran en las gráficas.
Data.java	Agrupar los diferentes elementos de datos.
Legend.java	Leyendas formadas por un rectángulo y la descripción.
XAxis.java	Eje de coordenadas X.
XYGraph.java	Construye la gráfica incorporando todos los elementos (ejes, leyendas, datos, etc).
XYGraphPanel.java	Panel sobre el que se construyen la gráfica.
YAxis.java	Eje de coordenadas Y.

CAPÍTULO 5. FUNCIONAMIENTO

A nivel de funcionamiento, el simulador comienza por una fase de inicialización de parámetros. A continuación, se muestra la pantalla principal, que tiene el siguiente aspecto:

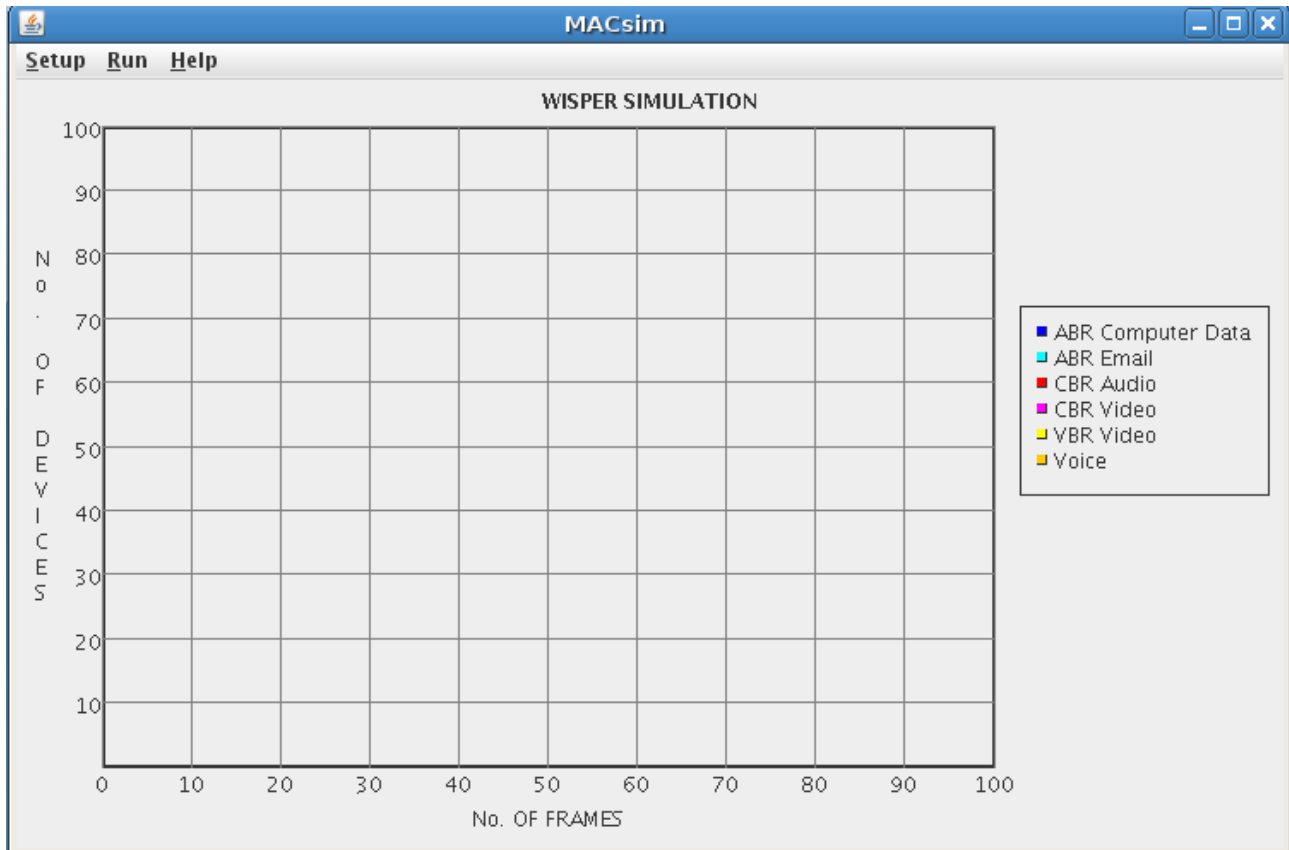


Fig. 5.1 Pantalla principal del simulador MACsim

Como puede observar en la figura anterior, la aplicación dispone de tres menús principales:

- Setup

Este es el menú de configuración. Permite configurar la utilidad de traza, la de estadísticas, las tramas, las ranuras, el gestor de tráfico y los distintos tipos de tráfico (datos ABR, email ABR, audio CBR, vídeo CBR, vídeo VBR y voz).

- Run

Este menú permite iniciar y cancelar la simulación mediante las opciones Start y Stop.

- Help

Este menú ofrece la opción de visualizar la información acerca de la aplicación (nombre, versión, descripción, etc.) y la licencia con la que se ha liberado (GPL versión 3).

5.1 Menú Setup

Este menú se utiliza para configurar la aplicación MACsim a partir de las siguientes opciones:

Logger

La primera opción del menú Setup permite configurar la utilidad de traza, mediante el siguiente cuadro de diálogo:

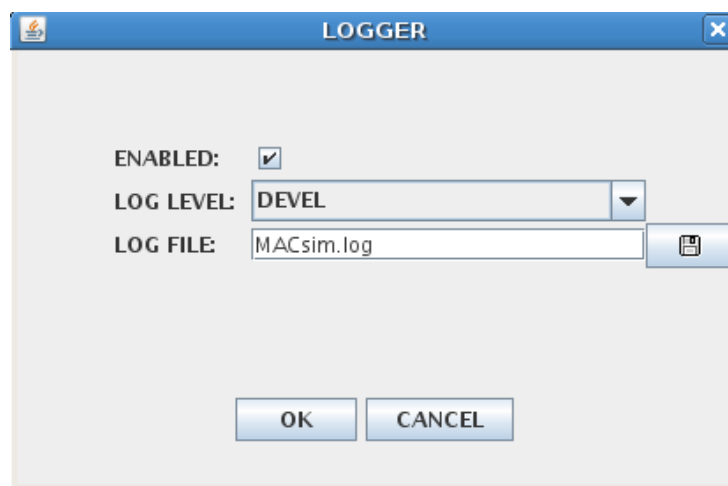


Fig. 5.2 Diálogo de la utilidad de traza

Este cuadro de diálogo permite:

- Activar/Desactivar la utilidad de traza.
- Seleccionar el nivel de trazado (ERROR, WARN, INFO, DEBUG, DEVEL).
- Especificar el nombre del fichero de trazada.

Se recomienda desactivar esta utilidad si se realizan simulaciones generando un número elevado de dispositivos, ya que hace un uso intensivo de la entrada/salida a disco que ralentiza bastante la simulación.

Statistics

La segunda opción del menú Setup permite configurar la utilidad de estadísticas, mediante el siguiente cuadro de diálogo:

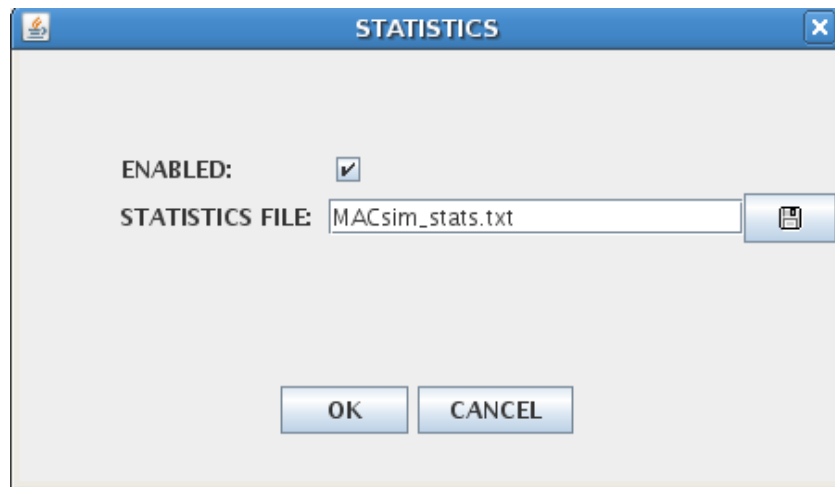


Fig. 5.3 Diálogo de la utilidad de estadísticas

Este cuadro de diálogo permite:

- Activar/Desactivar la utilidad de estadísticas.
- Especificar el nombre del fichero de estadísticas.

Frame

La tercera opción del menú Setup permite configurar las tramas, mediante el siguiente cuadro de diálogo:

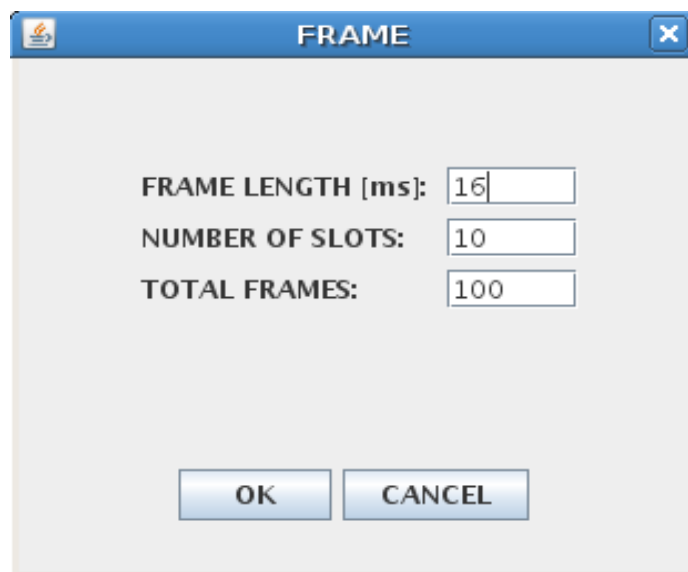


Fig. 5.4 Diálogo de configuración de tramas

Este cuadro de diálogo permite:

- Especificar la longitud de las tramas en milisegundos.
- Especificar el número de ranuras por trama.
- Especificar el número total de tramas.

Slot

La cuarta opción del menú Setup permite configurar las ranuras, mediante el siguiente cuadro de diálogo:

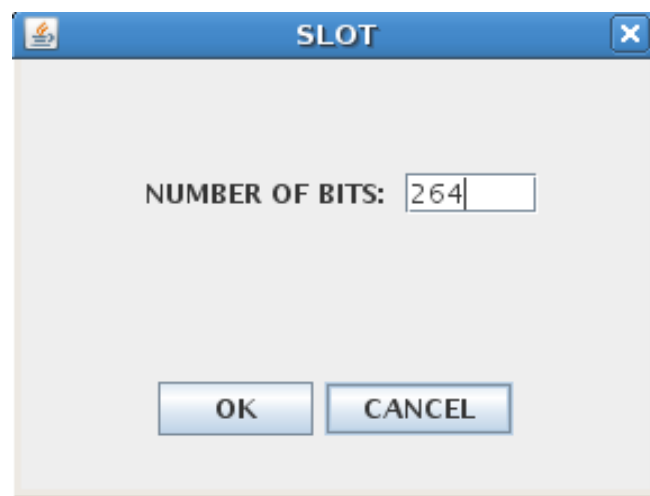


Fig. 5.5 Diálogo de configuración de ranuras

Este cuadro de diálogo permite:

- Especificar el número de bits por ranura.

Traffic

La quinta opción del menú Setup permite acceder al submenú para configurar el gestor de tráfico y los tipos de tráfico ha simular. Consta de las siguientes opciones:

- General
- ABR Computer Data
- ABR Email
- CBR Audio
- CBR Video
- VBR Video
- Voice

General

La primera opción del submenú Traffic permite configurar el gestor de tráfico, mediante el siguiente cuadro de diálogo:

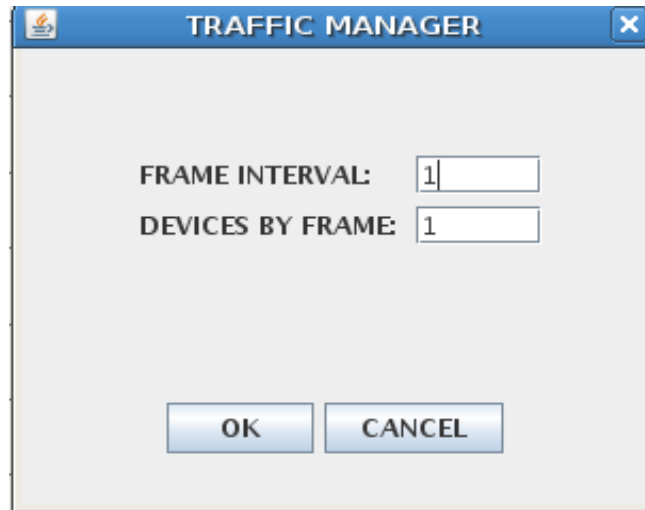


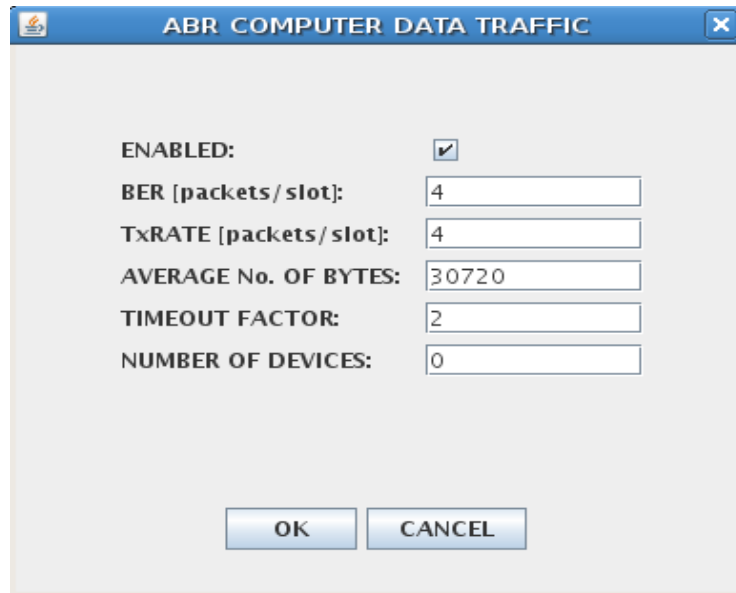
Fig. 5.6 Diálogo de configuración del gestor de tráfico

Este cuadro de diálogo permite:

- Especificar el intervalo de generación de dispositivos en tramas.
- Especificar el número de dispositivos a generar en dicha trama.

ABR Computer Data

La segunda opción del submenú Traffic permite configurar el tipo de tráfico Datos ABR, mediante el siguiente cuadro de diálogo:



ENABLED:	<input checked="" type="checkbox"/>
BER [packets/slot]:	<input type="text" value="4"/>
TxRATE [packets/slot]:	<input type="text" value="4"/>
AVERAGE No. OF BYTES:	<input type="text" value="30720"/>
TIMEOUT FACTOR:	<input type="text" value="2"/>
NUMBER OF DEVICES:	<input type="text" value="0"/>

OK CANCEL

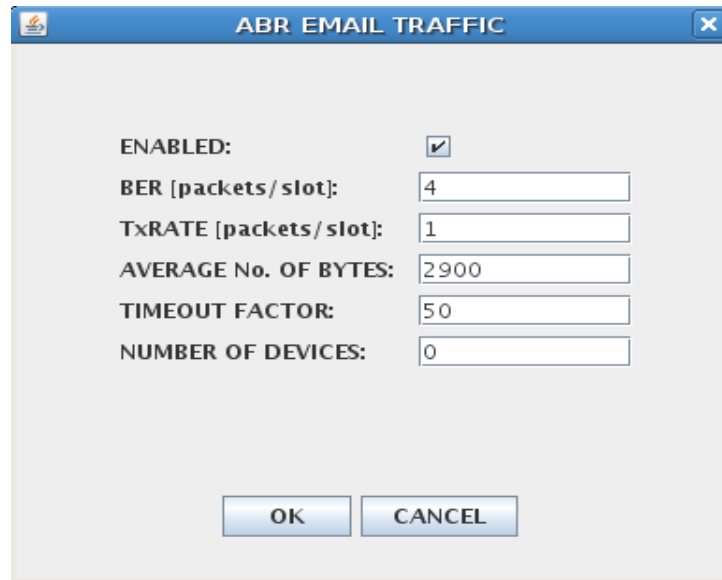
Fig. 5.7 Diálogo de configuración del tipo tráfico Datos ABR

Este cuadro de diálogo permite:

- Activar/desactivar este tipo de tráfico.
- Especificar el BER (Bit Error Rate) en paquetes por ranura.
- Especificar la tasa de transmisión en paquetes por ranura.
- Especificar el número medio de paquetes a transmitir.
- Especificar el factor de ponderación del timeout.
- Especificar el número de dispositivos a generar.

ABR Email

La tercera opción del submenú Traffic permite configurar el tipo de tráfico Email ABR, mediante el siguiente cuadro de diálogo:



ENABLED:	<input checked="" type="checkbox"/>
BER [packets / slot]:	<input type="text" value="4"/>
TxRATE [packets / slot]:	<input type="text" value="1"/>
AVERAGE No. OF BYTES:	<input type="text" value="2900"/>
TIMEOUT FACTOR:	<input type="text" value="50"/>
NUMBER OF DEVICES:	<input type="text" value="0"/>

OK CANCEL

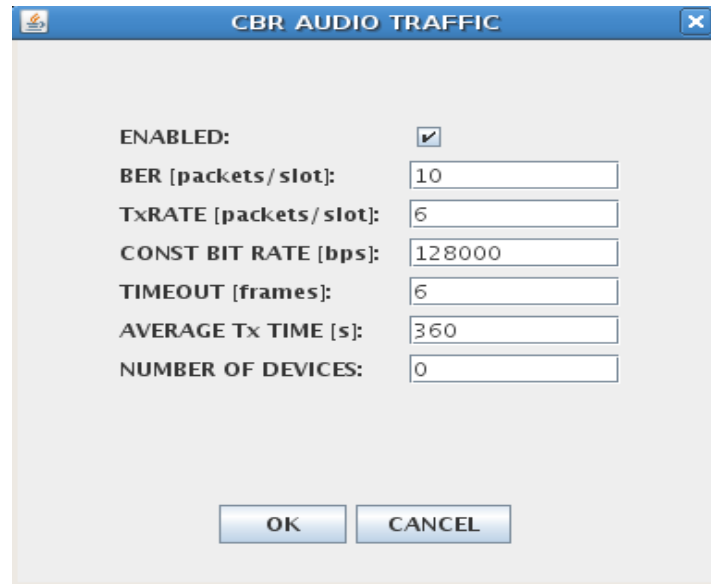
Fig. 5.8 Diálogo de configuración del tipo de tráfico Email ABR

Este cuadro de diálogo permite:

- Activar/desactivar este tipo de tráfico.
- Especificar el BER (Bit Error Rate) en paquetes por ranura.
- Especificar la tasa de transmisión en paquetes por ranura.
- Especificar el número medio de paquetes a transmitir.
- Especificar el factor de ponderación del timeout.
- Especificar el número de dispositivos a generar.

CBR Audio

La cuarta opción del submenú Traffic permite configurar el tipo de tráfico Audio CBR, mediante el siguiente cuadro de diálogo:



The image shows a dialog box titled "CBR AUDIO TRAFFIC" with a close button (X) in the top right corner. The dialog contains several configuration options, each with a label and a corresponding input field or checkbox:

Label	Value
ENABLED:	<input checked="" type="checkbox"/>
BER [packets/slot]:	10
TxRATE [packets/slot]:	6
CONST BIT RATE [bps]:	128000
TIMEOUT [frames]:	6
AVERAGE Tx TIME [s]:	360
NUMBER OF DEVICES:	0

At the bottom of the dialog, there are two buttons: "OK" and "CANCEL".

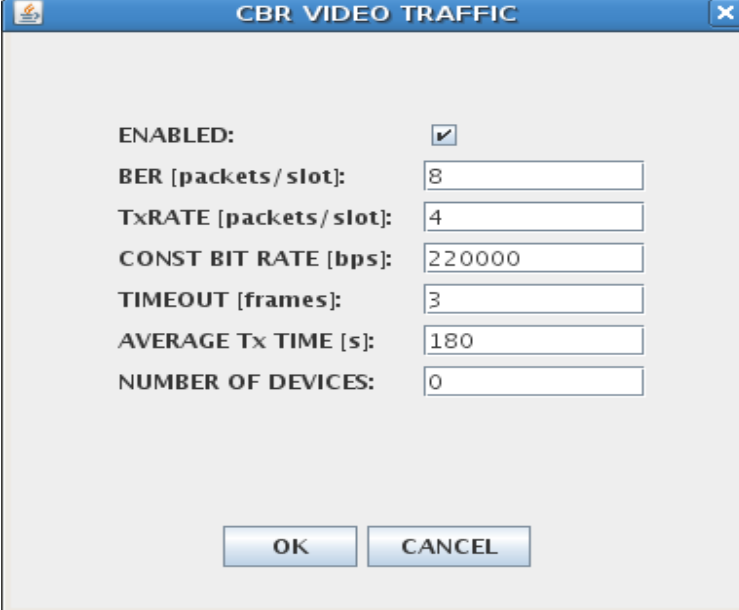
Fig. 5.9 Diálogo de configuración del tipo de tráfico Audio CBR

Este cuadro de diálogo permite:

- Activar/desactivar este tipo de tráfico.
- Especificar el BER (Bit Error Rate) en paquetes por ranura.
- Especificar la tasa de transmisión en paquetes por ranura.
- Especificar la tasa constante de bits que genera la fuente en bits por segundo.
- Especificar el timeout en número de tramas.
- Especificar el tiempo de transmisión medio en segundos.
- Especificar el número de dispositivos a generar.

CBR Video

La quinta opción del submenú Traffic permite configurar el tipo de tráfico Vídeo CBR, mediante el siguiente cuadro de diálogo:



The image shows a dialog box titled "CBR VIDEO TRAFFIC" with a close button (X) in the top right corner. The dialog contains several configuration options, each with a label and a corresponding input field or checkbox:

- ENABLED:
- BER [packets / slot]:
- TxRATE [packets / slot]:
- CONST BIT RATE [bps]:
- TIMEOUT [frames]:
- AVERAGE Tx TIME [s]:
- NUMBER OF DEVICES:

At the bottom of the dialog, there are two buttons: "OK" and "CANCEL".

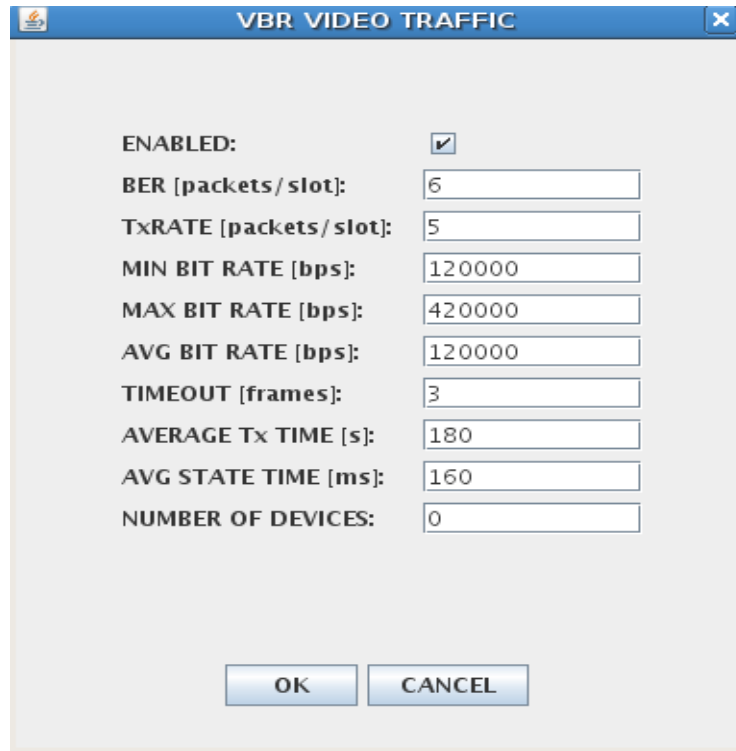
Fig. 5.10 Diálogo de configuración del tipo de tráfico Vídeo CBR

Este cuadro de diálogo permite:

- Activar/desactivar este tipo de tráfico.
- Especificar el BER (Bit Error Rate) en paquetes por ranura.
- Especificar la tasa de transmisión en paquetes por ranura.
- Especificar la tasa constante de bits que genera la fuente en bits por segundo.
- Especificar el timeout en número de tramas.
- Especificar el tiempo de transmisión medio en segundos.
- Especificar el número de dispositivos a generar.

VBR Video

La sexta opción del submenú Traffic permite configurar el tipo de tráfico Vídeo VBR, mediante el siguiente cuadro de diálogo:



ENABLED:	<input checked="" type="checkbox"/>
BER [packets / slot]:	<input type="text" value="6"/>
TxRATE [packets / slot]:	<input type="text" value="5"/>
MIN BIT RATE [bps]:	<input type="text" value="120000"/>
MAX BIT RATE [bps]:	<input type="text" value="420000"/>
AVG BIT RATE [bps]:	<input type="text" value="120000"/>
TIMEOUT [frames]:	<input type="text" value="3"/>
AVERAGE Tx TIME [s]:	<input type="text" value="180"/>
AVG STATE TIME [ms]:	<input type="text" value="160"/>
NUMBER OF DEVICES:	<input type="text" value="0"/>

OK CANCEL

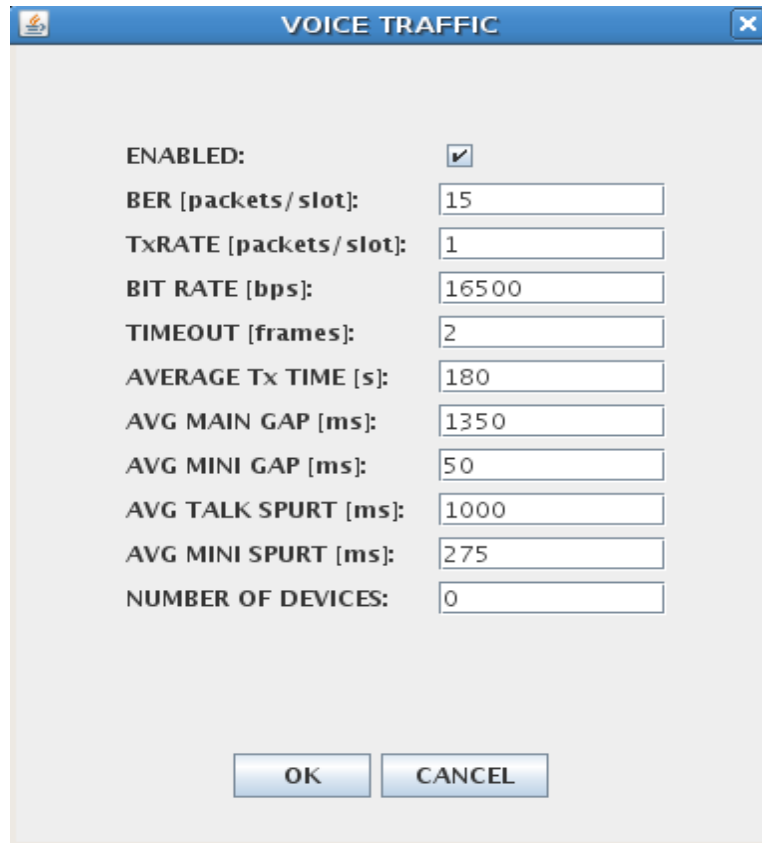
Fig. 5.11 Diálogo de configuración del tipo de tráfico Vídeo VBR

Este cuadro de diálogo permite:

- Activar/desactivar este tipo de tráfico.
- Especificar el BER (Bit Error Rate) en paquetes por ranura.
- Especificar la tasa de transmisión en paquetes por ranura.
- Especificar la tasa mínima de bits que genera la fuente en bits por segundo.
- Especificar la tasa máxima de bits que genera la fuente en bits por segundo.
- Especificar la tasa media de bits que genera la fuente en bits por segundo.
- Especificar el timeout en número de tramas.
- Especificar el tiempo de transmisión medio en segundos.
- Especificar la duración media de un estado en milisegundos.
- Especificar el número de dispositivos a generar.

Voice

La séptima y última opción del submenú Traffic permite configurar el tipo de tráfico Voz, mediante el siguiente cuadro de diálogo:



ENABLED:	<input checked="" type="checkbox"/>
BER [packets/ slot]:	<input type="text" value="15"/>
TxRATE [packets/ slot]:	<input type="text" value="1"/>
BIT RATE [bps]:	<input type="text" value="16500"/>
TIMEOUT [frames]:	<input type="text" value="2"/>
AVERAGE Tx TIME [s]:	<input type="text" value="180"/>
AVG MAIN GAP [ms]:	<input type="text" value="1350"/>
AVG MINI GAP [ms]:	<input type="text" value="50"/>
AVG TALK SPURT [ms]:	<input type="text" value="1000"/>
AVG MINI SPURT [ms]:	<input type="text" value="275"/>
NUMBER OF DEVICES:	<input type="text" value="0"/>

OK CANCEL

Fig. 5.12 Diálogo de configuración del tipo de tráfico Voz

Este cuadro de diálogo permite:

- Activar/desactivar este tipo de tráfico.
- Especificar el BER (Bit Error Rate) en paquetes por ranura.
- Especificar la tasa de transmisión en paquetes por ranura.
- Especificar la tasa de bits que genera la fuente en bits por segundo.
- Especificar el timeout en número de tramas.
- Especificar el tiempo de transmisión medio en segundos.
- Especificar la duración media del estado de silencio principal en milisegundos.
- Especificar la duración media del estado de mini silencio en milisegundos.
- Especificar la duración media del estado de habla en milisegundos.
- Especificar la duración media del estado de habla espúreo en milisegundos.
- Especificar el número de dispositivos a generar.

5.2 Menú Run

Este menú se utiliza para iniciar o parar la simulación mediante las siguientes opciones:

Run

La primera opción del menú Run permite iniciar la simulación. Empiezan a generarse tramas en función del valor especificado en la configuración. Para cada trama, si aplica, se generan dispositivos móviles en forma de peticiones de transmisión y se almacenan en la cola de peticiones TxRequestQueue. A continuación, se mapea la cola de peticiones en la cola con registros. Existe un registro para cada tipo de tráfico y dentro del registro hay una serie de grupos de paquetes pertenecientes a un mismo terminal móvil. Una vez la cola de registros está actualizada, se lanzan los procedimientos de acomodación y asignación de paquetes en las ranuras. Finalmente, se actualizan las colas de peticiones y de registros, borrando aquellas peticiones o grupos de paquetes que han expirado o están vacíos. Durante este proceso se actualizan las estadísticas con los paquetes entregados, los paquetes perdidos y el tiempo de entrega de los paquetes.

Stop

La segunda opción del menú Run permite finalizar la simulación, dejando la gráfica generada hasta el momento en pantalla. Hay que destacar que si se detiene la simulación no se generan estadísticas.

5.3 Menú Help

Este menú se utiliza para mostrar la información de la aplicación mediante la opción About.

About

La primera y única opción del menú Help muestra la información de la aplicación y su licencia mediante los siguientes cuadros de diálogo:



Fig. 5.13 Diálogo About

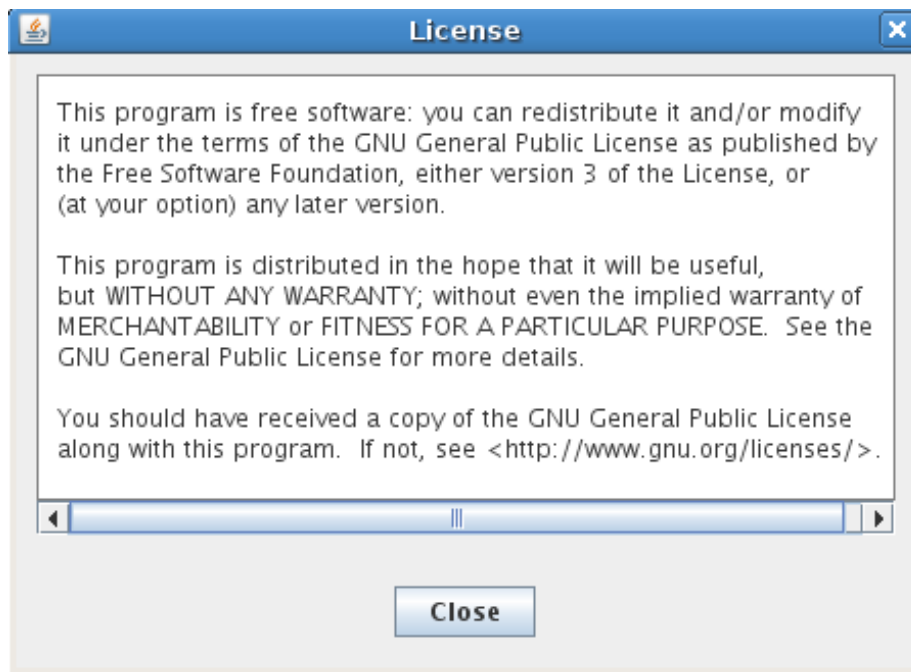


Fig. 5.14 Diálogo License

CAPÍTULO 6. LIBERACIÓN

A la hora de liberar la aplicación MACsim a la comunidad de software libre se han seguido los siguientes pasos:

- Elección de la licencia.
- Documentación del código fuente mediante javadoc.
- Preparación del paquete MACsim_v1.0.tar.gz.
- Alta de usuario en SourceForge.
- Alta del proyecto en SourceForge.
- Liberación del paquete MACsim_v1.0.tar.gz

6.1. Elección de la licencia

Como se ha comentado con anterioridad, la licencia elegida ha sido la GPL (GNU Public License) versión 3. Esta licencia permite la distribución binaria y del código fuente y realizar modificaciones sin restricciones o integrarlo en otras aplicaciones siempre que el código resultante tenga una licencia GPL o compatible. Por lo tanto, asegura que el código licenciado bajo ella siempre será libre. La versión 3 también aporta cláusulas de defensa contra las patentes de software, resuelve ambigüedades de la versión anterior y facilita su adaptación a otros países.

Se ha añadido una copia de la licencia entera en el fichero COPYING del paquete liberado, un extracto en cada uno de los ficheros fuente y en el diálogo License de la propia aplicación.

6.2. Documentación del código fuente

Todas las clases generadas constan de documentación en formato HTML, realizado mediante la herramienta javadoc. Este es el sistema de documentación estándar de las aplicaciones desarrolladas en Java. Puede encontrarse una copia de dicha documentación en el directorio doc del paquete liberado. El objetivo es facilitar que cualquier persona pueda continuar con el desarrollo de la aplicación.

6.3. Preparación del paquete MACsim_v1.0.tar.gz

Se ha generado el paquete MACsim_v1.0.tar.gz. Para instalarlo, basta con descomprimir y desempaquetarlo en el directorio deseado:

```
$ tar xvfz MACsim_v1.0.tar.gz
```

Este comando genera el directorio MACsim, dentro del cual se encuentran los siguientes directorios y ficheros:

- Directorio bin
Contiene los ficheros .class
- Directorio doc
Contiene la documentación de las clases generadas en formato HTML.
- Directorio images
Contiene las imágenes que se utilizan en la aplicación.
- Directorio log
Es donde se almacenan los ficheros de log.
- Directorio src
Contiene todo el código fuente.
- Directorio stats
Es donde se almacenan los ficheros de estadísticas.
- Fichero COPYING
Es la copia de la licencia GPL versión 3.
- Fichero MACsim
Es el script que lanza la ejecución del simulador MACsim.

Para ejecutar la aplicación, basta con lanzar el script MACsim (siempre que se tenga el JRE - Java Runtime Environment - en el path):

```
$ ./MACsim
```

Por el momento, la aplicación sólo está disponible para entornos GNU/Linux, aunque cualquier usuario puede crear su propia versión para Windows, adaptando el script anterior.

6.4. Alta de usuario en SourceForge

El alta de usuario en SourceForge es un requisito para poder dar de alta el proyecto en este portal.

6.5. Alta del proyecto en SourceForge

Una vez obtenida la cuenta de usuario en SourceForge, se ha dado de alta el proyecto MACsim. La página principal del proyecto es la siguiente:

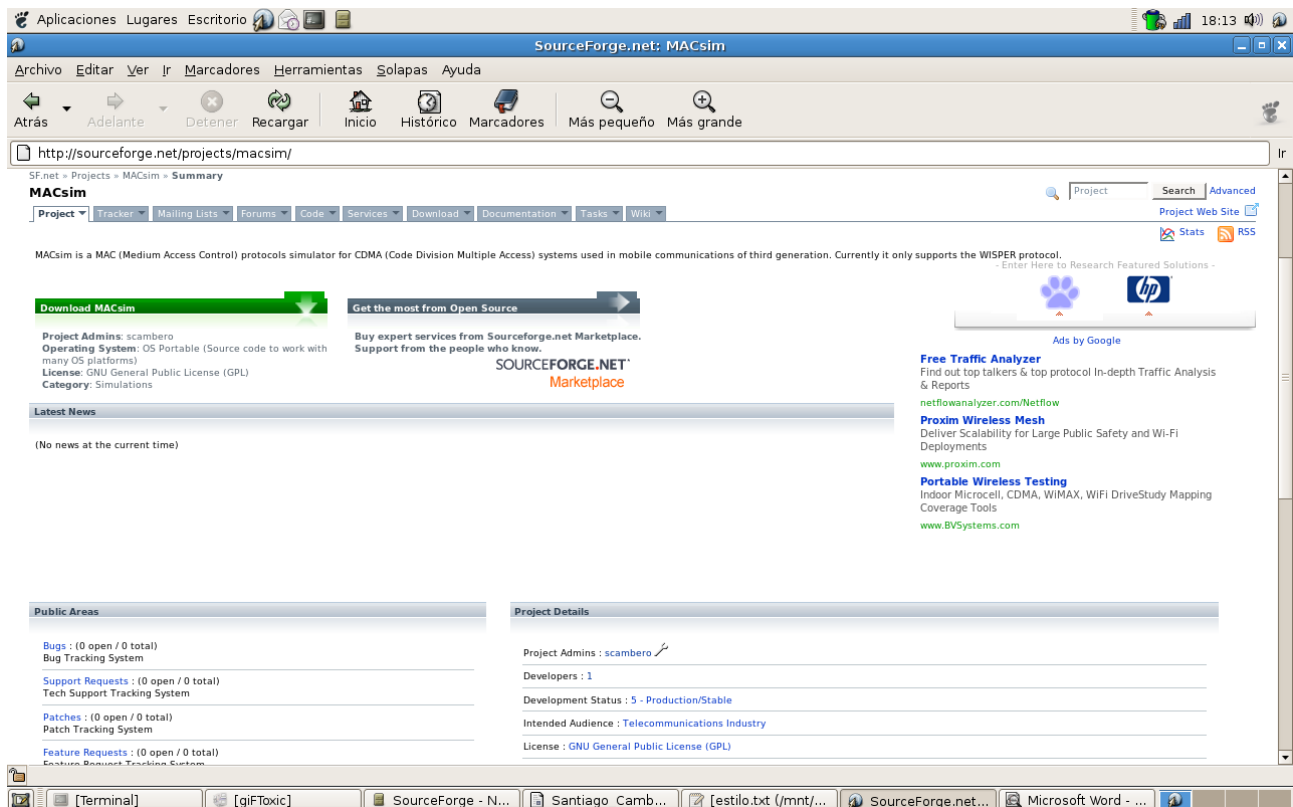
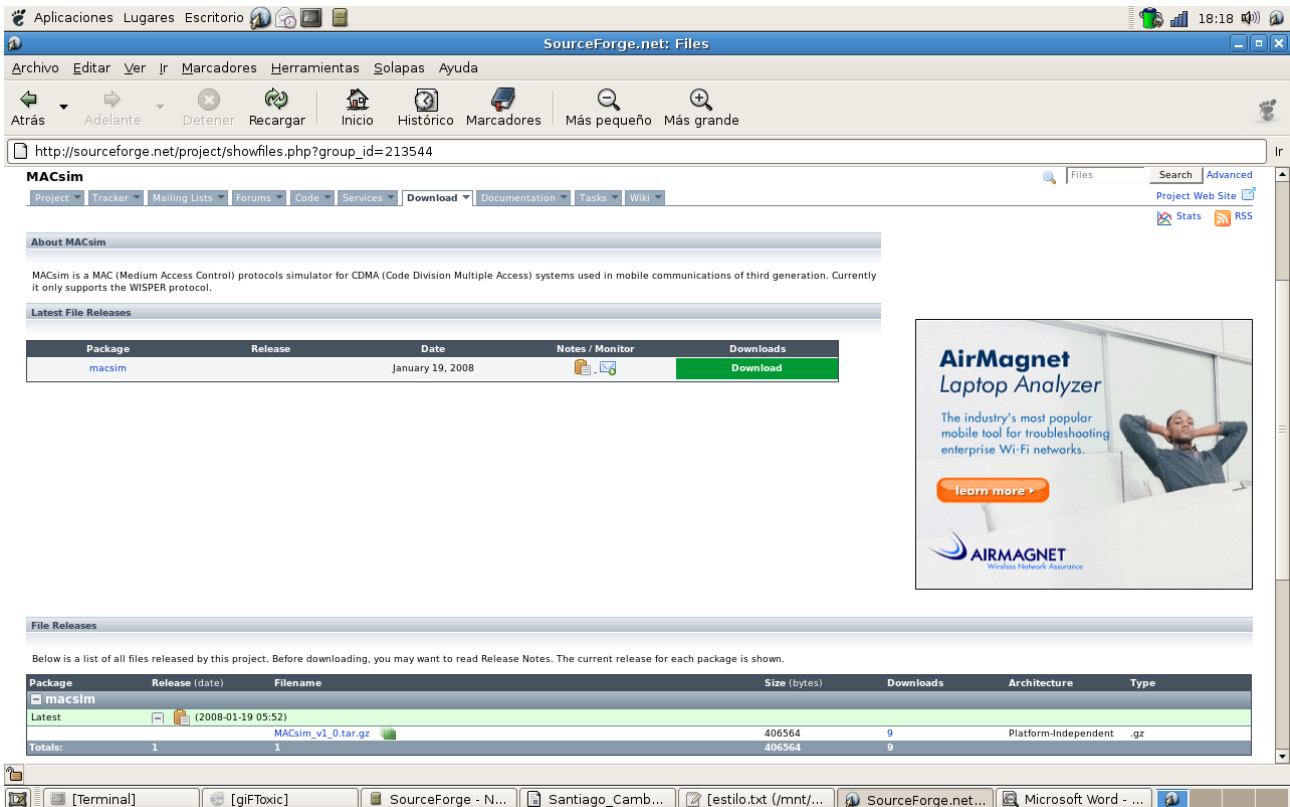


Fig 6.1 Página principal del proyecto MACsim en SourceForge

6.6. Liberación del paquete MACsim_v1.0.tar.gz

La aplicación esta disponible para descargarla mediante el enlace Download MACsim de la página principal. La página de descarga es la siguiente:



The screenshot shows the SourceForge project page for MACsim. The browser window title is "SourceForge.net: Files". The address bar shows the URL: http://sourceforge.net/project/showfiles.php?group_id=213544. The page content includes:

- About MACsim:** A brief description stating that MACsim is a MAC (Medium Access Control) protocols simulator for CDMA (Code Division Multiple Access) systems used in mobile communications of third generation, currently supporting the WISPER protocol.
- Latest File Releases:** A table with the following data:

Package	Release	Date	Notes / Monitor	Downloads
macsim		January 19, 2008		Download
- File Releases:** A section with a table listing all files released by the project. The current release is highlighted in green.

Package	Release (date)	Filename	Size (bytes)	Downloads	Architecture	Type
macsim						
Latest	(2008-01-19 05:52)	MACsim_v1_0.tar.gz	406564	9	Platform-independent	.gz
Totals:	1	1	406564	9		

An advertisement for AirMagnet Laptop Analyzer is visible on the right side of the page. The taskbar at the bottom shows several open applications: Terminal, gifToxic, SourceForge - N..., Santiago_Camb..., testilo.txt (/mnt/..., SourceForge.net..., and Microsoft Word - ...

Fig. 6.2 Página de descarga del proyecto en SourceForge

CAPÍTULO 7. CONCLUSIONES

En el presente trabajo se ha diseñado, implementado y liberado una aplicación de software libre denominada MACsim para evaluar protocolos MAC. Concretamente, la aplicación simula el protocolo WISPER uno de los múltiples protocolos aparecidos en la literatura para sistemas de comunicaciones móviles de tercera generación.

En el primer capítulo se han expuesto las motivaciones que han impulsado a realizar este proyecto, así como los objetivos que se pretendían conseguir inicialmente y la metodología a seguir.

En el segundo capítulo se ha realizado la planificación del proyecto. En base a los objetivos a alcanzar se han identificado una serie de tareas, tanto teóricas como prácticas, y se han planificado en función de la previsión temporal de cada una de ellas.

En el tercer capítulo, se ha presentado el protocolo de forma teórica. Se ha justificado la necesidad de protocolos MAC que soporten tráfico multimedia, bien asignando un umbral de potencia para cada tipo de tráfico transmitido o bien, planificando las transmisiones en las distintas ranuras en función del requerimiento BER. WISPER utiliza este último método, por lo que se puede seguir usando el control clásico de potencia. Los objetivos básicos, como son maximizar el rendimiento y minimizar las pérdidas, se consiguen planificando la transmisión en función del tipo de tráfico, el número de paquetes y el tiempo de expiración.

Se ha descrito la estructura de tramas, con la presencia de ranuras de petición, control y datos, en el enlace ascendente y en el descendente y el decalado temporal que debe haber entre ellas. También, se han descrito las expresiones para calcular el valor de prioridad de los paquetes, así como el número de paquetes a transmitir y los algoritmos de acomodación y asignación. Se ha propuesto un ejemplo para clarificar el funcionamiento del protocolo. Finalmente, se han identificado y descrito los tipos de tráfico presentes en los entornos de comunicaciones móviles multimedia.

En el cuarto capítulo, se ha expuesto el diseño y la implementación de la aplicación. En base a las necesidades del simulador, se ha propuesto una implementación en lenguaje Java con un cierto número de clases y se han descrito cada una de ellas.

Aunque este capítulo no sea especialmente extenso, ha sido el punto del trabajo que se ha llevado la mayor parte del tiempo. Implementar y sobretodo, depurar el código ha sido bastante costoso, ya que WISPER utiliza una exhaustiva manipulación de datos (gestión de estructuras de cola, cálculos de prioridades y número de paquetes con permiso de transmisión, asignaciones de paquetes en función de condiciones especiales). Precisamente, este exhaustivo tratamiento de datos es uno de los inconvenientes principales de este protocolo. Aunque no

es especialmente complejo, a nivel de programación se traduce en un gran número de objetos, funciones, variables y sentencias de control (bucles y condicionales), que a la hora de depurar implica tiempo y trabajo. Adicionalmente al núcleo de WISPER (protocolos de asignación y acomodación de paquetes), se han tenido que desarrollar los distintos tipos de tráfico, la interfaz gráfica y un módulo gráfico para visualizar las simulaciones en tiempo real. La implementación de estos tres elementos también ha requerido una cantidad de tiempo superior a la inicialmente planificada.

En el quinto capítulo se ha expuesto el funcionamiento del simulador, así como las distintas pantallas y el significado de todos los elementos que permiten configurar y personalizar las simulaciones. Es posible configurar desde los distintos tipos de tráfico, las tramas, las ranuras hasta las utilidades auxiliares de traza y estadísticas.

En el sexto capítulo se han descrito los pasos seguidos para liberar la aplicación, desde la elección de la licencia hasta el alta del proyecto en SourceForge, pasando por el método de instalación y ejecución del paquete generado.

En cuanto a los objetivos, se ha conseguido desarrollar una aplicación de software libre que simule el funcionamiento del protocolo WISPER. La aplicación consta de una interfaz gráfica donde es posible configurar todos los parámetros de la simulación, se muestra la evolución de las simulaciones en tiempo real y es posible obtener estadísticas de ellas. Únicamente, por falta de tiempo, no se han implementado los módulos de exportación de estadísticas a formato CVS y XML, así como su almacenamiento en una base de datos MySQL.

El proceso de liberación ha sido el objetivo que se ha visto perjudicado por los retrasos en la fase de implementación. Se ha elegido la licencia, desarrollado la aplicación en un entorno de software libre (Eclipse), se ha documentado a nivel de desarrollador, se ha empaquetado y se ha hospedado en el portal SourceForge. Sin embargo, no se ha podido subir el código al sistema de control de versiones, no se ha podido configurar la herramienta de seguimiento de fallos ni la lista de correo, no se ha documentado a nivel de usuario ni se ha podido hacer publicidad del proyecto. Esto ha originado que no haya habido ninguna respuesta por parte de la comunidad, a parte de nueve descargas en poco más de un día desde la página de SourceForge.

Los motivos de estos objetivos no alcanzados han sido una planificación excesivamente optimista en la fase de implementación, que ha originado que el resto de tareas se hallan visto retrasadas en el tiempo. También hay que considerar que el proyecto era bastante ambicioso en cuanto a la aplicación a desarrollar. Quizás hubiese sido preferible recortar funcionalidades (interfaz gráfica, gráficas en tiempo real, tipos de tráfico). Sin embargo, no implementar las partes más vistosas de la aplicación es de suponer que también habría tenido un impacto negativo en la respuesta de la comunidad.

De cara al futuro, sería interesante integrar nuevos protocolos en la aplicación. Para sistemas CDMA hay protocolos como MDPBMA-BB, que se utilizan en el enlace ascendente de UTRA en UMTS y utiliza un algoritmo que planifica las

transmisiones en función de la mejor relación retardo-rendimiento. Otro protocolo interesante es WCDMA, utilizado en el estándar IMT-2000. Permite múltiples servicios simultáneos en el terminal móvil y gestiona los recursos mediante una serie de formatos de transmisión variables. Este protocolo es probablemente el más complejo de todos ellos.

Otros aspectos a mejorar serían añadir una ayuda en el menú Help, implementar símbolos en las gráficas, exportar las estadísticas a CVS, XML y bases de datos y permitir guardar la configuración de forma persistente en fichero.

REFERENCIAS BIBLIOGRÁFICAS

I. F. Akyildiz, D. A. Levine, and I. Joe, "A slotted CDMA protocol with BER scheduling for wireless multimedia networks," *IEEE/ACM Trans. Net.*, vol. 7, no. 2, Apr. 1999, pp. 146–59.

I. F. Akyildiz, J. McNair, L. C. Martorell, R. Puigjaner and Y. Yesha, "Medium Access Control Protocols for Multimedia Traffic in Wireless Networks", *IEEE Network*, Jul./Aug. 1999, pp. 39–47.

F. Bauchot et al., "MASCARA, a MAC protocol for wireless ATM," *Proc. ACTS Mobile Summit '96*, Granada, Spain, Nov. 1996, pp. 647–51.

G. Bianchi et al., "CPRMA: A centralized packet multiple access for local wireless communications," *IEEE Trans. Vehic. Tech.*, vol. 46, May 1997.

A. E. Brand and H. Aghvami, "Multidimensional PRMA with prioritized Bayesian broadcast: A MAC strategy for multiservice traffic over UMTS," *IEEE Trans. Veh. Technol.*, vol. 47, pp. 1148–1161, Nov. 1998.

E. Dahlman et al., "UMTS/IMT-2000 based on wideband CDMA," *IEEE Commun. Mag.*, vol. 36, Sept. 1998, pp. 70–80.

D. A. Dyson and Z. J. Haas, "The dynamic packet reservation multiple access scheme for multimedia traffic," *ACM/Baltzer J. Mobile Networks & Apps.*, 1999.

K. S. Gilhousen, I. M. Jacobs, R. Padovani, A. J. Viterbi, L. A. Weaver, and C. E. Wheatley III, "On the capacity of a cellular CDMA system," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 303–312, May 1991.

D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthim, "Packet reservation multiple access for local wireless communications," *IEEE Trans. Commun.*, vol. 37, pp. 885–890, Aug. 1989.

D. J. Goodman and S. X. Wei, "Efficiency of packet reservation multiple access," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 170–176, Feb. 1991.

P. Jung, P. W. Baier, and A. Sreil, "Advantages of CDMA and spread spectrum techniques over FDMA and TDMA in cellular mobile applications," *IEEE Trans. Vehic. Tech.*, vol. 42, Aug. 1993, pp. 357–63.

Z. Liu, M. E. Zarki, M. J. Karol, and K. Y. Eng, "A demand-assignment access control for multi-code DS-CDMA wireless packet (ATM) networks," in *Proc. IEEE Int. Conf. Communications ICC'96*, June 1996.

R. Pichna and Q. Wang, "A medium access control protocol for a cellular packet CDMA carrying multirate traffic," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 1728–1736, Dec. 1996.

C. Roobol, P. Beming, J. Lundsjo, and M. Johanson, "A proposal for a MAC protocol for wideband CDMA capable of handling real-time and nonreal time services," in Proc. IEEE Veh. Technol. Conf., 1998.

A. Samukic, "UMTS universal mobile telecommunications system: Development of standard for the third generation," IEEE Trans. Vehic. Tech., vol. 47, Nov. 1998, pp. 1099–1104.

Z. Zhang and Y.-J. Liu, "Performance analysis of multiple access protocols for CDMA cellular and personal communication services," in Proc. IEEE INFOCOM'93, vol. 3, pp. 1214–1221, 1993.

H. Zhang, "Service disciplines for guaranteed performance service packet-switching networks," Proc. IEEE, vol. 83, pp. 1374–1399, Oct. 1995.

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify

you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal

- authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
 - D. Preserve all the copyright notices of the Document.
 - E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
 - F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
 - G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
 - H. Include an unaltered copy of this License.
 - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
 - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
 - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
 - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
 - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
 - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
 - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.