



**Escola Politècnica Superior  
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TREBALL DE FI DE CARRERA

**TÍTOL DEL TFC: Software de gestión en redes móviles**

**TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica**

**AUTOR: Jordi Collell i Codina**

**DIRECTOR: Carlos M. Ramírez Casañas**

**DATA: 22 de febrer de 2006**

**Título:** Software de gestión en redes móviles

**Autor:** Jordi Collell Codina

**Director:** Carlos M. Ramírez Casañas

**Data:** 22 de febrer de 2006

**Resum:**

Este documento es un estudio que tiene como finalidad maximizar el número de slots (canales) consecutivos vacíos en las frames (tramas) GPRS / EDGE, teniendo en cuenta que éstas transportan tanto tráfico de datos como tráfico de voz.

Se divide en dos partes:

- La primera presenta una introducción básica a la tecnología móvil celular sobre la que trabajamos (EDGE) y un compendio de artículos que hablan de distintas políticas de asignación de canales para el tráfico de voz y de datos en las tramas.

- La segunda parte es fundamentalmente analítica y práctica. Trata de los modelos estudiados e implementados por los autores: MODELO 1: Estrategia de asignación de canales de voz de forma ordenada y MODELO 2: Estrategia de asignación de canales con recursos de reserva para tráfico de voz.

**Title:** Software about mobile networks management

**Author:** Jordi Collell Codina

**Director:** Carlos M. Ramírez Casañas

**Date:** February, 22nd 2006

**Overview:**

This document is a study that it proposes to maximize consecutive free time slots number in GPRS/EDGE frames considering that these transport data traffic and voice traffic.

It is divided in two parts:

- First, it is theoretical and it gathers a basic introduction to the cellular mobile technology we work on (EDGE) and an article compendium that speaks about different allocation policies of times slots for voice and data channels.
- Second, it is basically analytical and practical. It deals with the models studied and implemented by the authors: *MODEL 1: Strategy of ordered voice channels allocation* and *MODEL 2: Strategy of channels allocation with resource reservation for voice traffic*.

# INDICE

<b>1. INTRODUCCIÓN</b> .....	<b>1</b>
<b>2. TECNOLOGÍA MÓVIL CELULAR: EDGE</b> .....	<b>2</b>
<b>3. IMPLEMENTACIÓN DE SOFTWARE PARA EL ANÁLISIS Y CÁLCULO DE RÁFAGAS</b> .....	<b>7</b>
3.1. Gestión de recursos radio.....	7
3.2. Descripción de la metodología .....	9
3.3. Diagramas y notación de los modelos.....	10
<b>4. MODELO 1: Estrategia de asignación de canales de voz de forma ordenada</b> .....	<b>11</b>
4.1. Modelo teórico .....	11
4.2. Implementación del modelo.....	13
<b>5. MODELO 2: Estrategia de asignación de canales con recursos de reserva para tráfico de voz</b> .....	<b>16</b>
5.1. Modelo teórico .....	16
5.2. Implementación del modelo.....	24
<b>6. ANÁLISIS DE PROBABILIDADES DE ENCONTRAR RÁFAGAS</b> .....	<b>31</b>
6.1. Implementación .....	31
<b>7. RESULTADOS</b> .....	<b>33</b>
7.1. General .....	34
7.1.1 Caso 1 .....	34
7.2. Número de slots fijo vs. $PB_{total}$ variable .....	34
7.2.1. Caso 1 .....	34
7.2.2. Caso 2 .....	35
7.2.3. Caso 3 .....	36
7.2.4. Comentario.....	37
7.3. Número de slots variable vs. $PB_{total}$ fijo .....	38
7.3.1 Caso 1 .....	38
7.3.2 Caso 2 .....	39
7.3.3. Caso 3.....	40
7.3.4. Comentario.....	41
7.4. Número de slots variable vs. $PB_{total}$ variable .....	42
7.4.1. Caso 1 .....	42
7.4.2. Caso 2 .....	43
7.4.3. Comentario.....	44
7.5 Conclusiones.....	45
<b>8. ESTUDIO DE AMBIENTALIZACIÓN</b> .....	<b>47</b>
<b>9. BIBLIOGRAFIA</b> .....	<b>48</b>
<b>10. ANEXOS: Parámetros de tráfico</b>	

# 1. INTRODUCCIÓN

Este TFC tiene como título “Software de gestión de redes móviles”. Así a simple vista este título puede resultar genérico del trabajo que se presenta en este trabajo fin de carrera. Además no se determina qué quiere gestionar ni a que tipo de redes móviles hace referencia. Esto ocurre, principalmente, ya que este TFC tiene la voluntad de formar parte de un amplio proyecto relacionado con lo que dice el título, la gestión de los recursos de las redes móviles. Así pues, el presente TFC se puede concretar y ubicar dentro de este amplio título.

En primer lugar, presentamos la tecnología que hemos usado para la realización del trabajo. Como especifica el título, nos basamos en las redes móviles celulares, y, concretamente en el funcionamiento básico de la tecnología EDGE/UMTS, de la cual se presentarán los aspectos más relevantes.

En segundo lugar, como la mayoría de los proyectos, este trabajo dispone de un objetivo general que nos planteamos cuando empezamos el TFC: Maximizar el número de slots (canales) consecutivos vacíos en los frames (tramas) GPRS / EDGE teniendo en cuenta que éstas transportan tanto tráfico de datos como tráfico de voz. Este mecanismo de asignación de canales es lo que se conoce como *Channel Allocation Strategy*, que no son más que las estrategias de asignación de canales (slots) en las correspondientes tramas.

Para llegar a conseguir este objetivo general nos planteamos objetivos específicos que detallamos a continuación:

- Decidir el modelo de análisis de la asignación de canales que queremos generar para poder analizarlo posteriormente.
- Generar el modelo y, a partir de éste, conseguir resultados congruentes.
- Contrastar los datos obtenidos según los dos objetivos anteriores, y trabajar con ellos para poder obtener una estrategia de asignación próxima a la más óptima posible.

Por un lado, también existe un objetivo más abstracto, relacionado con el espacio donde se ha desarrollado el trabajo: la Universidad. Y es que creemos que uno de los objetivos principales de la propia Universidad es aprender y adquirir conocimientos. Esto es lo que hemos intentado a partir de estudios anteriores sobre los aspectos tratados en este TFC, el trabajo realizado durante este tiempo y las conclusiones que se han obtenido.

Lo que presentamos a continuación no es más que un reflejo escrito de los pasos seguidos por nosotros desde septiembre de 2005 hasta febrero de 2006 para realizar este TFC.

## 2. TECNOLOGIA MOVIL CELULAR: EDGE

En general, EDGE es un estándar 3G aprobado por la ITU (*International Telecommunication Union*), y está respaldado por el Instituto Europeo de Estándares de Telecomunicaciones (ETSI), la Asociación de GSM (GSMA), el Proyecto Conjunto de Tercera Generación (3GPP) y 3G Américas.

EDGE es una tecnología radio en las redes móviles celulares que permite que las redes GSM ofrezcan servicios de tercera generación (3G) dentro de las frecuencias existentes. Como resultado evolutivo de GSM/GPRS, EDGE es una mejora a las redes GPRS y GSM. GPRS es una tecnología portadora de datos que EDGE refuerza con una mejora de la interfaz de radio, y proporciona una rápida conexión y velocidades de datos medios de 80-130Kbps y hasta 473kbps (tres veces mayores que las de GPRS). Añadir EDGE a la red de GPRS significa aprovechar en toda su extensión las redes de GSM.

Las principales diferencias técnicas entre EDGE y GPRS son:

**Tabla 2.1** – Diferencias entre GPRS y EDGE

	GPRS	EDGE
Modulation	GMSK	8-PSK/GMSK
Symbol rate	270 ksym/s	270 ksym/s
Modulation bit rate	270 kb/s	810 kb/s
Radio data rate per time slot	22,8 kb/s	69,2 kb/s
User data rate per time slot	20 kb/s (CS4)	59,2 kb/s (MCS9)
User data rate (8 time slots)	160 kb/s	473,6 kb/s
	(182,4 kb/s)	(553,6 kb/s)

Como podemos observar, aunque GPRS y EDGE tienen la misma *symbol rate*, la *modulation bit rate* difiere. EDGE puede transmitir tres veces más rápido que GPRS durante el mismo período de tiempo. Hay también diferencias entre las interfaces radio y las tasas de datos del usuario (*user data rates*) que son resultado de tener en consideración o no la cabecera del paquete. En función de esta consideración, las dos formas de calcular el *throughput* a menudo causan malentendidos en la industria con respecto al propio caudal entre EDGE y GPRS. La tasa de datos de 384kbps es a menudo usada por la ITU como el valor límite requerido para servicios (dentro del *estándar International Mobile Telecommunications-2000, IMT-2000*) en un entorno peatonal. Los 384 kbps se corresponden con los 48kbps por *time slot*, asumiendo ocho *time slots* por terminal.

Básicamente EDGE solo introduce respecto a GPRS una nueva técnica de modulación y un nuevo canal de codificación que se utiliza para transmitir mediante conmutación de circuitos y conmutación de paquetes para servicios de voz y datos. Se debe tener en cuenta que EDGE no puede funcionar sin GPRS, de hecho es una extensión de éste denominado EGPRS el que se emplea en EDGE.

Como hemos dicho anteriormente EDGE proporciona un *throughput* mayor que GPRS para cada usuario. Esto lo consigue ya que con EDGE el mismo *time slot* puede soportar más usuarios. Esto decrementa el número de recursos radio requeridos para soportar el mismo tráfico, y así se libera capacidad para más servicios de datos o voz. EDGE consigue así hacer más fácil la coexistencia de conmutación de circuitos y conmutación de paquetes. En redes planificadas con el espectro limitado, EDGE también puede ser visto como un elevador de capacidad para el tráfico de datos.

EDGE se puede desplegar en las bandas de frecuencia 800, 900, 1800 y 1900 MHz actuales y puede servir como la vía a la tecnología UMTS (WCDMA).

UMTS son las siglas que en inglés hace referencia a los Servicios Universales de Telecomunicaciones Móviles. De la misma forma UTRAN (*UMTS Radio Access Network*) es el sistema de acceso radioeléctrico de UMTS que soporta las operaciones con una alta eficiencia espectral y calidad de servicio.

EDGE complementa a UMTS y así ambas tecnologías coexisten. EDGE y UMTS comparten la misma red central IP (Protocolo de Internet) y el Proyecto Conjunto de Tercera Generación (3GPP) las apoya. EDGE se ubica en el espectro existente y como complemento a los despliegues de UMTS ya que al igual que ésta, EDGE es una solución que interopera con los despliegues de 3G de EDGE.

Los operarios de TDMA tienen la opción de desplegar GSM/GPRS/EDGE sobreponiéndola de manera paralela a sus redes de TDMA tanto en 850MHz, como en 1900MHz. En un mercado de 850MHz (enlace a la tabla del equipo de 850MHz), especialmente en áreas menos pobladas y rurales, EDGE ofrece una excelente calidad de propagación; cuanto más baja sea la posición en el espectro, tomará menos energía al enviar la señal y ésta viajará a mayor distancia.

Hace falta preguntarnos el porqué de este interés en hacer convivir UMTS y EDGE. Las razones son claras:

- **Mayor cobertura 3G:** EDGE complementa a UMTS de la misma manera en que GPRS complementa a EDGE. Una importante ventaja del camino evolutivo a 3G basado en GSM es que los clientes conmutan automáticamente entre las redes dependiendo de factores tales como la cobertura. Los fabricantes han incluido EDGE en todos los teléfonos y dispositivos UMTS. Esto significa que cuando los clientes se desplazan fuera de la cobertura UMTS,

sus dispositivos automáticamente conmutan a EDGE, garantizando la continuidad de servicio 3G.

- **Flexibilidad:** EDGE proporciona al operador la flexibilidad de lanzar la infraestructura de tercera generación por etapas. Los costes de desplegar UMTS son mayores, de manera que es sensato restringir UMTS a áreas que concentran la mayor cantidad de clientes potenciales, mientras que EDGE brinda 3G en ciudades más pequeñas y zonas rurales.
- **Coste:** Los operadores tienen una justificación de negocio para elegir EDGE en virtud de que, por sólo 2 € por cliente en actualizaciones, pueden ofrecer servicio 3G. EDGE no es un paso interino en vano: tanto EDGE como UMTS reutilizan la infraestructura de paquetes de datos desplegada para GPRS.
- **Performance (Rendimiento):** No todas las aplicaciones de datos inalámbricas requieren las velocidades que ofrece UMTS, de modo que encaminar cada aplicación por la red apropiada asegura el mejor uso posible del espectro.

El estándar GSM/EDGE ha evolucionado con el fin de poder conseguir un acceso 3G de calidad. Con este objetivo 3GPP ha ido publicando actualizaciones (*releases*). Además, para simplificar la interoperatividad entre redes GSM/EDGE y UMTS se decidió que la arquitectura de sus respectivas redes de acceso fuese aproximadamente la misma, siendo la única diferencia la tecnología de la interfaz radio. Siguiendo esta idea en el *release 5* se especificó GERAN (*GSM/EDGE Radio Access Network*), para hacer explícito que se trata de una arquitectura de red nueva.

En este *release* se definieron cuatro clases de tráfico distintas que son iguales que las de UMTS: *conversational*, *streaming*, *interactive* y *background*.

En la clase *conversational* la voz o el video es transportada entre usuarios finales. Los requerimientos de QoS son principalmente determinados por la percepción humana. Si fracasa se puede producir una degradación que puede llevar a un nivel inaceptable de QoS.

El tráfico de *streaming* es un tráfico de datos del usuario en una dirección sin ninguna exigencia de retardo absoluta. Sin embargo este tráfico es importante para preservar la relación de tiempo entre entidades de información. El *stream* se puede alinear en tiempo en el receptor mediante el uso de un buffer de datos. Durante la transmisión se acepta un cierto retardo en la fase.

El tráfico interactivo consiste en sesiones donde un humano o bien una máquina piden datos a una máquina remota. En este tráfico es característico un cuadro de preguntas-respuestas transferidas.



El tráfico de *background* tiene la posibilidad de retardo más relajada de las cuatro clases de tráfico. El tráfico de datos se puede hacer en background cuando el retraso no es un problema. Por ejemplo en el tráfico de SMS.

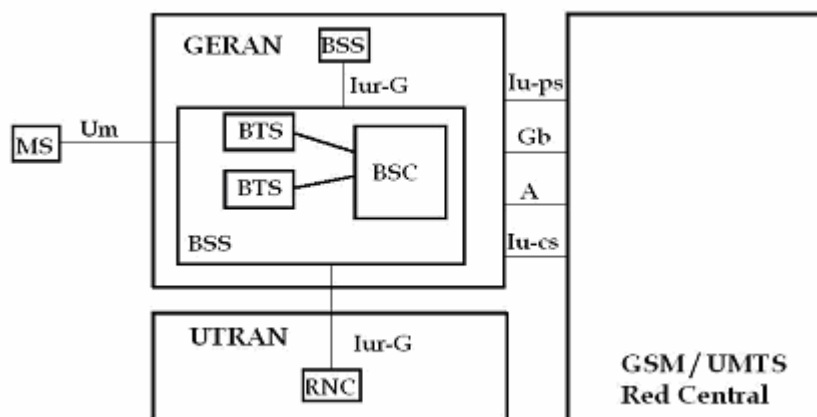
Refiriéndonos a la arquitectura de GERAN, decir que el soporte para los servicios de paquetes en tiempo real y la adopción de la arquitectura de Calidad de Servicio (QoS) de WCDMA, requiere cambios en la red central GPRS de segunda generación. En lugar de introducir estos cambios, otra solución atractiva es conectar GERAN a la red central WCDMA/GPRS de tercera generación, la cual soporta los servicios en tiempo real y la arquitectura de QoS de WCDMA. Esto permite una red central común para UTRAN y GERAN, conectados sobre una interfaz común (ver [6]).

Los principios básicos usados para diseñar la arquitectura GERAN fueron:

- Separar las funcionalidades radio y las no radio entre la red central y la red de acceso de radio. Esto permite el desarrollo de la plataforma que permite el uso de servicios independientes para cada tipo de acceso.
- Asegurar el servicio para terminales anteriores al release 5
- Maximizar las cosas comunes entre GERAN y UTRAN pero manteniendo la compatibilidad anterior para GERAN. El punto de encuentro entre GERAN i UTRAN es la interfaz que los une a la red central: *Iu*. Es de esta forma que GERAN puede ofrecer los mismos servicios que UTRAN.

Como hemos dicho en el párrafo anterior, para conectarse a la red central WCDMA/GPRS de tercera generación, GERAN utiliza la interfaz *Iu*. La interfaz *Iu* está compuesta de dos partes: la *Iu-ps*, que se conecta con el dominio de conmutación de paquetes de la red central y la *Iu-cs*, que se conecta con el dominio de conmutación de circuitos de la red central.

La figura muestra que GERAN también se conecta con los nodos de la red central de segunda generación.



**Fig. 2.1** – Arquitectura GERAN

Las interfaces *A* y *Gb* permanecen intactas respecto a *releases* anteriores, haciendo que GERAN sea totalmente compatible para soportar terminales de las versiones anteriores de GSM, GPRS, ECSD y EGPRS. La razón principal para soportar las interfaces *A* y *Gb* con las versiones anteriores y no utilizar la interfaz *lu-ps* para las terminales nuevas, es que la separación funcional entre la red de acceso radioeléctrico y la red central difiere substancialmente entre *lu* y *A/Gb*.

La interfaz *Gb* es usada en GSM/GPRS para conectar el conocido como *Serving GPRS Support Node* (SGSN) y el Subsistema de la Estación Base (BSS).

Por otro lado, la interfaz *A* es la interfaz usada tradicionalmente en GSM para conectar la BSS y el centro de conmutaciones móviles (MSC).

La interfaz de radio entre la estación móvil (MS) y GERAN, llamada la interfaz *Um*, está basada en la interfaz del enlace de radio anterior. Sin embargo, se han especificado varias mejoras sobre diferentes capas de protocolos de enlace de radio, para proveer portadores de radio adecuados para los servicios en tiempo real.

Ejemplos de estas mejoras son el soporte para la reelección de la celda para el dominio de conmutación de paquetes, la separación de los planos de usuario y de control, y los modos transparentes en las capas de protocolo del enlace radio. También será posible multiplexar tráfico de paquetes de datos hacia y desde las terminales operando ya sea en los modos *lu* o *Gb*, sobre el mismo intervalo de tiempo.

Existe también la interfaz *lur-g* que proporciona una conexión lógica entre dos GERANs. Está basada en la interfaz *lur* de UTRAN (ver [5]).

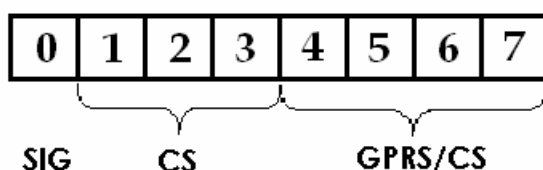
Con los *releases* nuevos [6] se ha conseguido que GERAN provea un rango completo de servicios inalámbricos de tercera generación. También está incluido el soporte para todas las clases de servicio especificadas para WCDMA, con los requerimientos de tiempo real de la clase de servicio conversacional. Además, se consigue un alto nivel de alineación con WCDMA, por la interconexión hacia la red central WCDMA, sobre la interfaz *lu*, que es común con UTRAN. Además el transporte IP beneficia a los operadores, pues supone un descenso de los costes.

### 3. IMPLEMENTACIÓN DE SOFTWARE PARA EL ANÁLISIS Y CÁLCULO DE RÁFAGAS

#### 3.1. Gestión de recursos radio

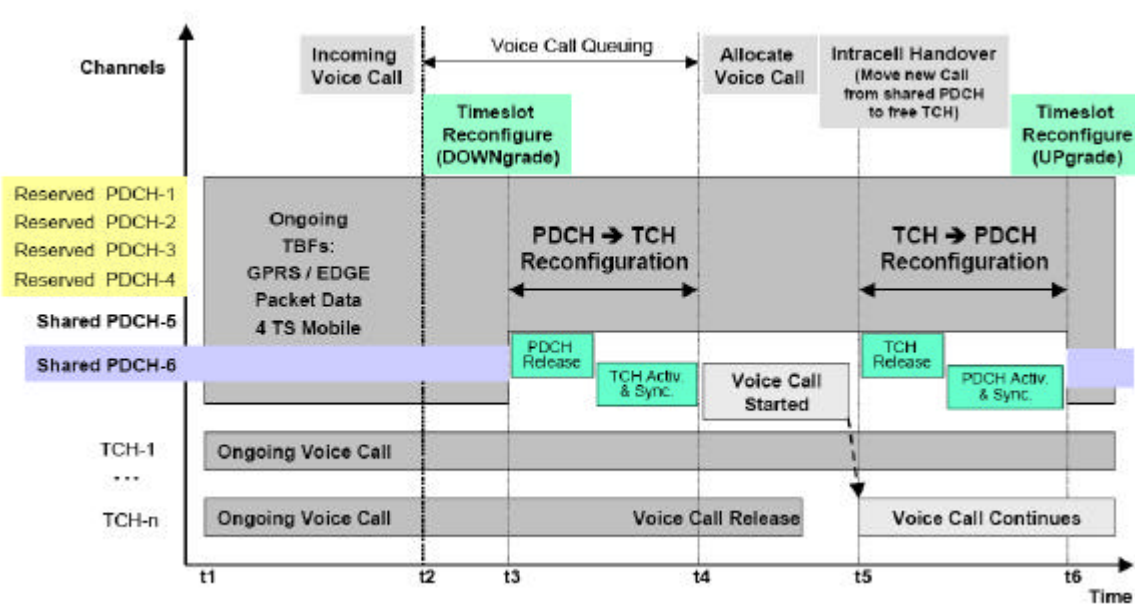
El dimensionado de redes móviles es un tema ampliamente estudiado y es posible encontrar un número considerable de artículos que hacen referencia a aspectos englobados en el dimensionado de estas redes. Después de haber estudiado una selección de los artículos más relevantes, nos hemos fijado en algunos aspectos que están contrastados y que nos permitieran diseñar un modelo que nos sirviese para conseguir nuestros objetivos.

Los autores Jarkko Itkonen y Jukka Lempiäinen en [8] analizan la capacidad de la interfaz aire para conmutación de circuitos y paquetes basándose en la comparación de los resultados teóricos y experimentales de la pérdida de celda en una red GSM. Es realmente interesante como se reparten los slots en las celdas estudiadas: se reservan unos slots solo para voz (CS), uno para señalización (SIG) y los demás se comparten (GPRS/CS). Se observa en la siguiente imagen:



**Fig. 3.1** – Repartición de slots en el modelo de Itkonen y Lempiäinen

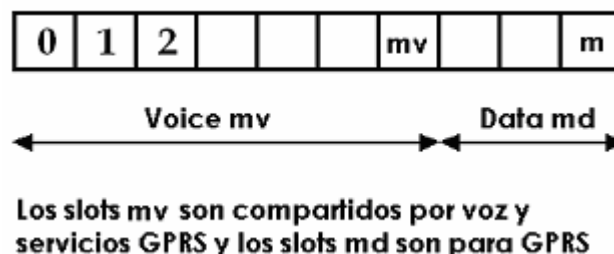
El artículo presentado por K. Ivanov, C.F. Ball y F. Trembl en [9] es interesante desde el punto de vista que presenta un modelo con tecnología GPRS/EDGE y plantean para cada trama canales compartidos para voz y datos (*PDCH*) y otros canales dedicados solo para voz (*TCH*). En el caso que llegue una llamada de voz en primer lugar ocupará los canales reservados. En el caso que estén todos los canales de voz ocupados uno de los canales compartidos se reconfigura para poder acoger la llamada de voz entrante. Es de resaltar que en el momento en el que quede libre un canal de voz de los reservados, se produce un traspaso en la misma celda (*intra-cell handover*) y la llamada de voz que estaba en uno de los slots compartidos ocupará el slot reservado que acaba de quedar libre. En la figura 3.2 se muestra el proceso de ocupación de los canales en el tiempo.



**Fig. 3.2** – Configuración de los slots de voz y datos y ejemplo de *intracell handover*.

Los autores no solo plantean el modelo sino realizan simulaciones con la asignación de los canales para distintos valores. Cabe destacar que, por bien que hay unos canales dedicados y compartidos, estos se llenan cuando quedan libres y de manera aleatoria. Además consideran una cola para las llamadas de voz en el supuesto caso que se tenga que hacer la reconfiguración *PDCH -> TCH*.

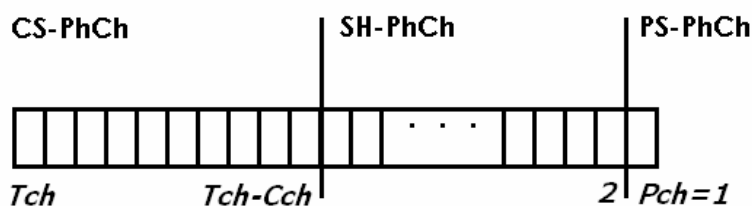
El modelo presentado por autores Shaoji Ni y Sven-Gustav Haggman en [10] se basa en gestionar todos los slots y separarlos de la siguiente forma: unos reservados solo para datos GPRS (*md*) y otros compartidos para voz y datos (*mv*). En el caso de los compartidos, las llamadas de voz tienen prioridad por delante de las de GPRS. Cuando llegue tráfico de voz y estén todos los slots ocupados, éste irá a una cola con disciplina FIFO. Estos autores proponen un modelo en el que el tráfico de voz, prioritario, obliga a ciertas transmisiones GPRS a pasar a un estado de suspendidas que estarán esperando en una cola más prioritaria que la cola de las nuevas entradas de datos GPRS.



**Fig. 3.3** - Repartición de slots en el modelo de Ni y Haggman

El de P. Fitzpatrick, M. Ivanovich y J. Yin en [11] es un modelo interesante, también: suponen una independencia entre llamadas de voz y tráfico de datos. Se estima la probabilidad de que una llamada de voz tenga prioridad respecto a una de datos y la probabilidad del tráfico de datos se retrase a causa de las llamadas de voz. De esta forma llegan a la conclusión que haciendo una zona de prioridad de voz y una de datos se obtienen resultados más eficientes.

En [12] Wessam Ajib y Philippe Godlewski plantean un modelo parecido al [8]. Hay slots reservados solo para voz (*CS-PhCh*) y slots para voz y datos (*SH-PhCh*). El número de slots reservado para voz depende de las llamadas de voz que lleguen. Reservan también un slot para un mínimo de QoS para datos (*PS-PhCh*).



**Fig. 3.4** - Repartición de slots en el modelo de Ajib y Godlewski

Wei-Yeh Chen, Jean-Lien C. Wu y Hung-Huan Liu en [13] muestran tres modelos. En el primero, se comparten todos los slots para voz y datos, con prioridad y posibilidad de robo de slot para el *handover*. En el segundo, emplea un buffer que ocupan los paquetes GPRS que son desplazados por las llamadas de voz, con mayor prioridad. Y el tercero, es el mismo modelo que el primero pero con un buffer para todos los paquetes de datos (tanto aquellos que quedan desplazados por voz como los que llegan y se encuentran todos los slots ocupados).

En último lugar estudiamos el documento interno de Carlos Ramírez y Josep Paradells [14]. Éste nos ha servido para explicar el método básico de asignación de canales del que queríamos empezar y que sirve de base a todo el trabajo realizado en este TFC.

## 3.2. Descripción de la metodología

Como ya se ha dicho anteriormente lo que queremos es trabajar con la asignación de slots.

Tal como vemos en [14], en los sistemas TDMA EDGE tenemos ocho slots por portadora, de los cuales uno es de señalización y los demás para voz o datos. Diferenciamos un slot de otro ya que cada slot tiene asignada una ID (un número de slot desde 0 a 7). Nosotros trabajamos pensando que los slots se asignan cogiendo el primer slot libre con la ID más baja. Un ejemplo es la siguiente imagen. Como podemos ver, en la figura 3.5 están los slots 0 y 2

ocupados. Cuando tenemos que asignar el siguiente slot se escoge el 1, pues es el que tiene la ID menor(2):



Fig. 3.5 – Ejemplo de ocupación de slots

Cuando un slot se libera significa que una llamada ha terminado. Independientemente del lugar que ocupa queda un hueco, pues no nos planteamos un proceso de *intra-cell handover* o reordenación de los slots.

### 3.3. Diagramas y notación de los modelos

Normalmente las transiciones entre estados de un sistema se representan mediante una Cadena de Markov, que es un grafo que presenta la evolución de un sistema. Sin embargo, estamos trabajando en unos modelos basados en tramas y slots. Es por esto que en lugar de seguir la notación tradicional en las cadenas de Markov usamos una notación en la que los estados son representados simulando una trama.

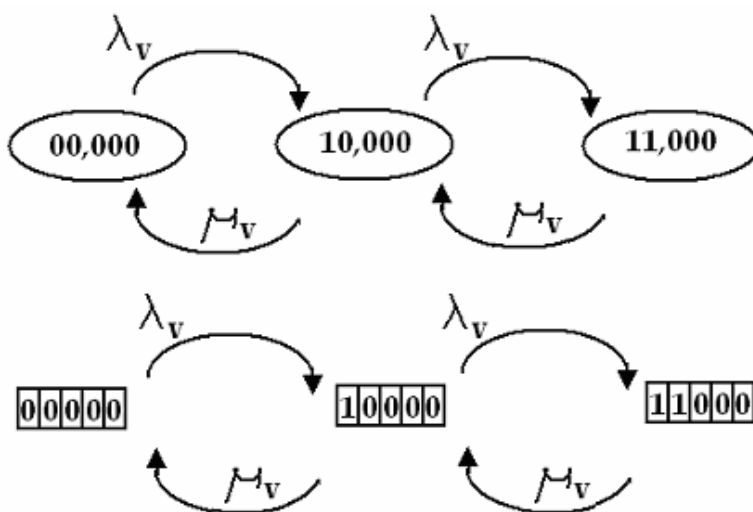


Fig. 3.6 – Cadena de Markov tradicional (arriba) y la usada en nuestro caso (abajo)

## 4. MODELO 1: Estrategia de asignación de canales de voz de forma ordenada

### 4.1. Modelo teórico

Empezamos con un modelo sencillo. Consideramos que solo llega tráfico de un tipo, voz. Cada llamada que llega ocupa el slot con ID menor, siguiendo la política explicada anteriormente. Esta característica (asignación secuencial) influye en el número de estados posibles que es sensiblemente mayor a los que habría en el caso en el que la asignación fuese aleatoria. Así pues, siguiendo el modelo presentado tenemos:

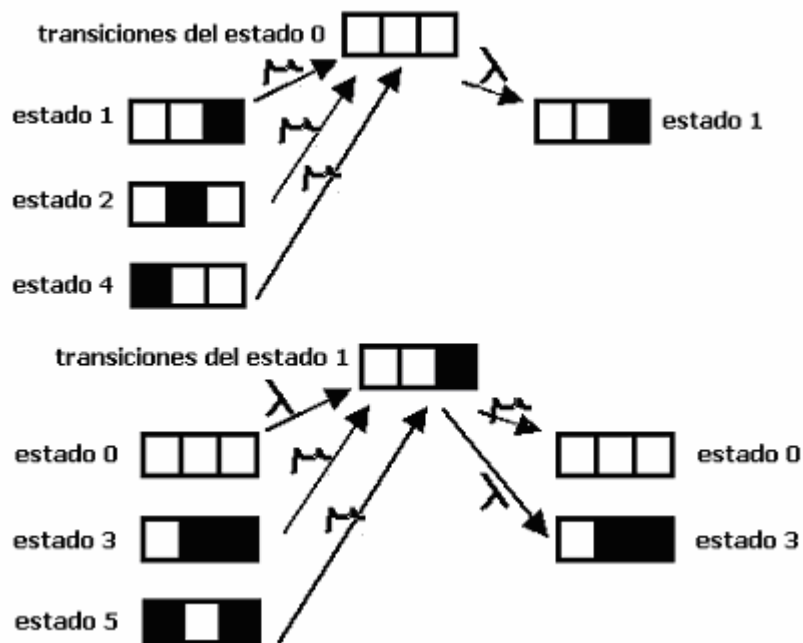
$$\text{Número de estados} = 2^{\text{num\_slots}} \quad (4.1)$$

Un problema que nos encontramos fue cómo numerar los estados. Lo solucionamos con lo que llamamos “pensar en binario”. La idea es simple. Los slots o están llenos o están vacíos, lo que equiparamos con están 0 (vacíos) o están 1 (llenos). Si observamos, la fórmula (4.1) si la aplicamos con tres slots veremos que tenemos ocho estados posibles. Si escribimos los estados en binario obtenemos:

0 -> 000
1 -> 001
2 -> 010
3 -> 011
4 -> 100
5 -> 101
6 -> 110
7 -> 111

Hemos seguido esta representación binaria para, dependiendo de qué slots estén llenos, definir en qué estados nos encontramos.

Una vez tenemos claro como llamamos a los distintos estados lo que tenemos que hacer, definimos las ecuaciones de transición. Para explicar como las conseguimos usamos un ejemplo con tres slots. En la figura 4.1 se muestran las transiciones correspondientes a los estados 0 y 1:



**Fig. 4.1** – Transiciones correspondientes a los estados 0 y 1

A partir de la figura 4.1, es fácil escribir las ecuaciones de transición de los estados 0 y 1.

$$\begin{aligned}
 \mathbf{1}P_0 - \mathbf{m}(P_1 + P_2 + P_4) &= 0 \\
 -\mathbf{1}P_0 + (\mathbf{1} + \mathbf{m})P_1 - \mathbf{m}(P_3 + P_5) &= 0
 \end{aligned}$$

De la misma forma obtenemos las ecuaciones de transición para todos los estados. Una vez las tenemos, construimos la matriz de transición de estado, que llamaremos *Matriz A*.

$$\text{Matriz A} = \begin{pmatrix}
 \mathbf{1} & -\mathbf{m} & -\mathbf{m} & 0 & -\mathbf{m} & 0 & 0 & 0 \\
 -\mathbf{1} & \mathbf{1} + \mathbf{m} & 0 & -\mathbf{m} & 0 & -\mathbf{m} & 0 & 0 \\
 0 & 0 & \mathbf{1} + \mathbf{m} & -\mathbf{m} & 0 & 0 & -\mathbf{m} & 0 \\
 0 & -\mathbf{1} & -\mathbf{1} & \mathbf{1} + 2\mathbf{m} & 0 & 0 & 0 & -\mathbf{m} \\
 0 & 0 & 0 & 0 & \mathbf{1} + \mathbf{m} & -\mathbf{m} & -\mathbf{m} & 0 \\
 0 & 0 & 0 & 0 & -\mathbf{1} & \mathbf{1} + 2\mathbf{m} & 0 & -\mathbf{m} \\
 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} + 2\mathbf{m} & -\mathbf{m} \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
 \end{pmatrix}$$



Donde cada fila indica los valores dependientes de cada ecuación de estado, excepto la última fila en la que sus valores son todos unos. Esta última fila es así ya que substituye todos los valores del estado siete, con la finalidad de provocar que la suma de probabilidades sea uno (condición indispensable). También definimos una matriz  $P$  tal que:

$$P = (P_0 \quad P_1 \quad P_2 \quad P_3 \quad P_4 \quad P_5 \quad P_6 \quad P_7)^T \quad (4.2)$$

la definimos la matriz de valores independientes  $B$  como:

$$B = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1)^T \quad (4.3)$$

con la finalidad de forzar que la suma de todas las probabilidades (recordemos que hemos sustituido la última fila por unos) sea la unidad.

Así la ecuación del sistema es  $A \cdot P = B$ . Si resolvemos este sistema de ecuaciones obtenemos las probabilidades de estar en cada estado.

## 4.2. Implementación del modelo

Partíamos de un *script* Matlab en el que, dependiendo del número de *slots* y la probabilidad de bloqueo deseada, se calculan las probabilidades de estar en cada estado. Para realizar dicho cálculo, consideramos un valor fijo de las tasa de llegada (?) del tráfico de voz, y haciendo uso de las tablas de Erlang encontramos el tiempo medio de llegadas ( $1/\mu$ ) correspondiente. El proceso de resolución del modelo sigue los siguientes pasos: en primer lugar crea la matriz  $A$ , posteriormente genera la  $B$  y, a partir de aquí, calcula las probabilidades.

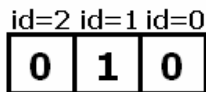
Partiendo del *script* inicial se asume la consideración de denominar los slots vacíos como '0' y los ocupados como '1'. Una función del *script* es crear una variable que toma todos los valores de los estados posibles, puesto que, como hemos visto anteriormente disponemos de una expresión mediante la cual podemos saber de antemano el número de estados para nuestro modelo.

```

N = 2^num_slots;
for fila=1:N
    ...
end
```

Mirando el manual de Matlab descubrimos una función que pasa de decimal a binario y viceversa [15]. El *script*, usando esta función, para cada valor decimal que toma la variable fila lo pasa a binario y lo almacena en un vector. Este vector es analizado campo por campo de la siguiente forma:

1. Cogemos un ejemplo con tres slots y la variable fila de valor 2. En decimal 2 es: 010.

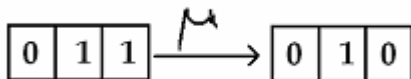


**Fig. 4.2**

2. Recorre el vector desde la ID más baja a la más alta. En el caso del ejemplo:

La posición 0 del vector es un '0'.

Podemos llegar a este estado con  $\mu$  desde el estado correspondiente a sustituir este '0' por un '1'



**Fig. 4.3**

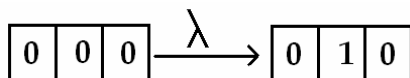
En la columna correspondiente al valor decimal del estado desde donde procedemos (en este caso '011', o sea 3) pone una  $-\mu$  (sabiendo que la numeración de las posiciones parte del valor 0 y, por tanto, para este caso será el cuarto valor que encontramos en la fila):

FILA 2 (0 0 0  $-\mu$  0 0 0 0)

Además existe un contador que tiene en cuenta si ha habido un '0' (lo usa para el caso que encuentre algún '1')

La posición 1 del vector es un '1'.

Esto quiere decir que podría ser posible llegar con  $\lambda$  a este estado desde el correspondiente estado cambiando el '1' por un '0'.



**Fig. 4.4**

Pero en este caso, esta transición no es posible, pues, si tenemos en cuenta la política que decidimos utilizar, vemos que hay un slot con ID mas baja vacío que sería el que se ocuparía en primer lugar. Este slot lo tenemos identificado con la variable que cuenta los '0' que anteriormente hemos aumentado.

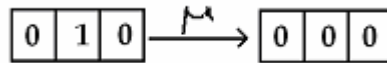


Fig. 4.5

Si la posición 2 del vector es un '0' ocurre lo mismo que con el slot con identificador 0.

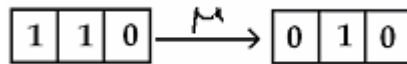


Fig. 4.6

En la columna correspondiente, valor decimal del estado de destino (en este caso '110', o sea 6), se inserta una  $-\mu$

$$\text{FILA 2 } (0 \ 0 \ 0 \ -m \ 0 \ 0 \ -m \ 0)$$

3. Para finalizar solo hay que tener en cuenta los estados a los que se puede ir desde el estado origen. En particular, no nos importan los valores de los estados, hemos observado que podemos ir con ? solo a un estado siempre que hay un '0' en alguno de los slots y que podemos ir con  $-\mu$  a tantos estados como '1' haya en el vector. Estos unos ya los hemos tenido en cuenta en el contador anteriormente citado, así en el mismo ejemplo que tratamos, la fila 2 quedaría:

$$\text{FILA 2 } (0 \ 0 \ 1 + m \ -m \ 0 \ 0 \ -m \ 0)$$

De esta forma se completaría para todos los valores y obtiene la matriz  $A$ . Una vez se obtiene  $A$ , se substituye la última fila por unos, se crea el vector  $B$  de la misma forma que hemos contado en la teoría y calcula la matriz  $P$  haciendo  $P = A \setminus B$ .

## 5. MODELO 2: Estrategia de asignación de canales con recursos de reserva para tráfico de voz

### 5.1. Modelo teórico

El modelo pensado se complica, pues introducimos tanto tráfico de datos como llamadas de voz. Está basado en el modelo presentado por [8] pero con la diferencia que nosotros lo planteamos haciendo una ocupación secuencial de los slots. De esta forma disponemos de slots reservados para llamadas de voz y el resto de los slots son compartidos por el tráfico de voz y el tráfico de datos (figura 5.1).

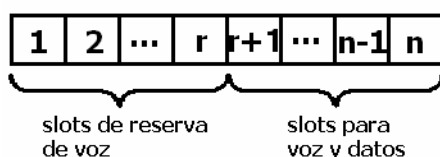


Fig. 5.1 – Asignación de slots en el modelo 2

En este caso llegan dos tipos de tráfico y, en consecuencia, con un método ordenado de asignación de los canales, tomamos la determinación de que las llamadas de voz lleguen por un lado (izquierda según la figura 5.2) y los datos por el lado contrario (derecha según la figura 5.2).

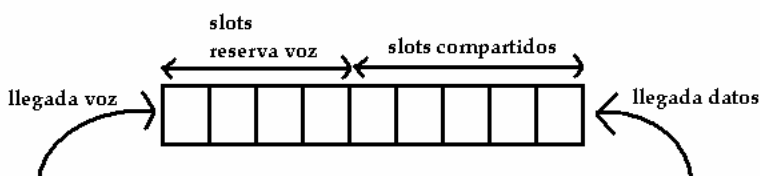


Fig. 5.2 – Llegada de tráfico en el modelo 2

Sin embargo, lo podemos presentar según la figura 5.3 que nos permite poder ver más claramente con qué modelo trabajamos.

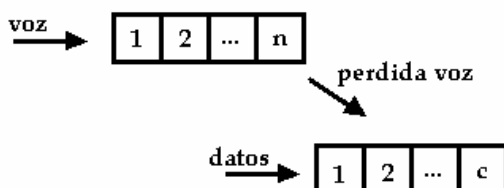


Fig. 5.3

Consideramos que las llamadas de voz llegan con una tasa  $I_v$  y se sirven con una tasa  $m_v$ . Las de datos llegan con una tasa  $I_d$  y se sirven con una tasa  $m_d$ . El campo llamada perdida de voz no es más que las llamadas de voz que llegan y se encuentran con todos los slots reservados para voz ocupados y, en consecuencia, se desbordan a los slots compartidos. Llamaremos a esta tasa  $I_{v-p}$ .

Basándonos en la teoría de Teletráfico los slots que están reservados solo para voz son un sistema de pérdidas puro (*Erlang B*). Si consideramos que:

$$A = I_v / m_v \tag{5.1}$$

Entonces siguiendo la teoría de un sistema de pérdidas puro obtenemos que:

$$I_{v-p} = I_v (E_{r_B}(A, n)) \tag{5.2}$$

donde  $n$  es el número de slots reservados para voz que hay en la trama. La ocupación de los slots la ejemplificamos para cuatro slots, de los cuales dos están reservados para voz y dos compartidos de la siguiente forma:

- Llega una llamada de voz y la alojamos en el primer slot vacío situado más a la izquierda:

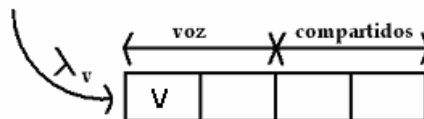


Fig. 5.4

- Llega una llamada de datos y la alojamos en el primer slot vacío situado más a la derecha:

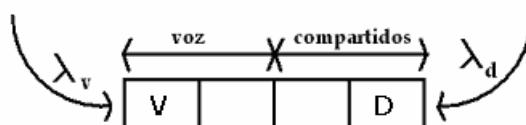


Fig. 5.5

- Llega otra llamada de voz. Se aloja siguiendo los criterios anteriormente comentados:

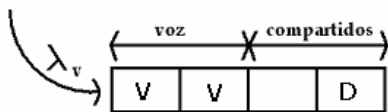


Fig. 5.6

- En el caso en el que estamos ahora podrían pasar dos cosas: la llegada de una llamada de voz o la llegada de tráfico de datos. Si llegase tráfico de datos éste lo haría con una tasa  $I_d$  y se alojaría en el slot que está vacío. Si llega una llamada de voz ésta lo haría con una tasa  $I_{v-p}$ :

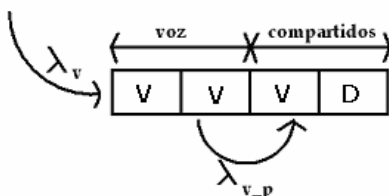


Fig. 5.7

A partir de este momento cualquier tráfico que llegue (independientemente que sea de datos o de voz) no se podrá atender y sería descartado.

Supongamos que se sirve una de las llamadas de voz alojadas en uno de los slots reservados exclusivamente para voz. Nosotros hemos planteado para estos casos un realojamiento automático de la llamada de voz situada más a la derecha de los slots compartidos (proceso de *intra-cell handover*). En el caso del ejemplo, si se desaloja un slot de voz quedaremos en este estado (figura 5.8)

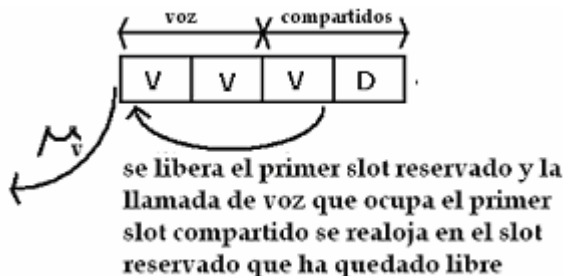


Fig. 5.8

y el estado resultante es el que muestra la figura 5.9.

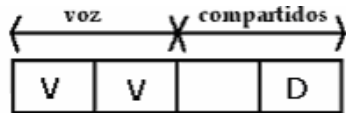


Fig. 5.9

Es importante resaltar en este modelo que nunca existirá la posibilidad de que haya uno de los slots reservados para voz vacíos y alguno de los compartidos cursando una llamada de voz. En el ejemplo que estamos tratando nunca ocurrirá una situación como esta (figura 5.10), pues se habría producido un realojamiento automático como el relatado anteriormente.

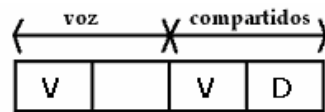


Fig. 5.10

Para nuestro estudio nos interesa saber el número de estados que tendremos. Este número varía según los slots de reserva de voz y los compartidos que tengamos. Después de estudiarlo obtuvimos la siguiente expresión:

$$\begin{aligned}
 N^{\circ} \text{ estados} &= 2^{\text{num\_slots} - \text{num\_reserva}} \cdot 2^{\text{num\_reserva}} \\
 &+ 3^{\text{num\_slots} - \text{num\_reserva}} - 2^{\text{num\_slots} - \text{num\_reserva}} \qquad (5.3)
 \end{aligned}$$

donde:

- **num\_slots** es el número total de slots, es decir, los reservados para voz más los compartidos.
- **num\_reserva** son los slots que reservamos para voz.

Analizando la expresión matemática,

- $2^{\text{num\_slots} - \text{num\_reserva}} \cdot 2^{\text{num\_reserva}}$  son todas las combinaciones posibles de voz y, para cada una de éstas, obtenemos las posibles combinaciones de solo datos en los slots compartidos.
- $3^{\text{num\_slots} - \text{num\_reserva}}$  representa las permutaciones con repetición expresadas por:  $PR^3_{\text{num\_slots} - \text{num\_reserva}}$ . Esto quiere decir las combinaciones posibles de 0, voz y datos teniendo en cuenta que

se pueden repetir los elementos (0, V, D) (de aquí el 3 puesto que son tres posibles estados en que se puede encontrar el slot).

- $-2^{num\_slots - num\_reserva}$  este término de la expresión aparece ya que, como en la parte anterior de la fórmula se hacen todas las combinaciones posibles, tenemos que restar aquellas que ya hemos tenido en cuenta anteriormente (primer término de la expresión) y que no son más que todas las combinaciones de voz en los slots reservados mientras los compartidos están vacíos.

Siguiendo el ejemplo anterior, aplicamos la fórmula para  $num\_reserva=2$  y  $num\_slots=4$  (recordar que teníamos 2 slots compartidos) y obtenemos que hay 21 estados ( $2 \cdot 2 + 9 - 4$ ). Si los escribimos (considerando slot vacío como '0', con voz 'V' y datos 'D') son:

0000	V000	0V00	VV00	000D	V00D
0V0D	VV0D	00D0	V0D0	0VD0	VVD0
00VV	D0VV	0VDD	VVDD	VVV0	VV0V
VVVD	VVDV	VVVV			

Otra cosa que nos interesa cuantificar es la probabilidad de bloqueo. Lo podemos hacer para datos y para voz. En el caso de tráfico de datos una petición de este tipo no se podrá atender siempre que estén todos los slots que compartan voz y datos ocupados. Esto será (siguiendo con el ejemplo anterior) para los siguientes casos:

00VV	D0VV	0VDD	VVDD	VVV0	VV0V
VVVD	VVDV	VVVV			

De forma general:

$$PB_{datos} = \sum_{s_1 + \dots + s_s = S} P_{v_1, \dots, v_{num\_reserva}, s_1, \dots, s_s} \quad (5.4)$$

donde  $S = num\_slots - num\_reserva$

Para voz consideramos dos probabilidades de bloqueo, la que sufren los slots reservados para tráfico de voz (independientemente de lo que pase en los slots compartidos) y la probabilidad de bloqueo total del sistema, que sería el caso en el que todos los slots se encontrasen ocupados.

Para el primer caso, en el ejemplo anterior se produciría uno de los bloqueos (slots reservados para tráfico de voz) en los siguientes estados:

VVDD	VVV0	VV0V	VV00	VV0D	VVD0
VVVD	VVDV	VVVV			



Que de forma general:

$$PB_{res\_voz} = \sum_{v_1 + \dots + v_{num\_reserva} = Vr} P_{v_1, \dots, v_{num\_reserva}, s_1, \dots, s_s} \quad (5.5)$$

donde  $Vr = num\_reserva$

Para el segundo caso, se produciría bloqueo siempre que todos los slots estuviesen ocupados, continuando con el ejemplo sería para los estados siguientes:

VVVD	VVDV	VVVV	VVDD
------	------	------	------

Que de forma general:

$$PB_{voz} = \sum_{v_1 + \dots + v_{num\_reserva} + s_1 + \dots + s_s = V} P_{v_1, \dots, v_{num\_reserva}, s_1, \dots, s_s} \quad (5.6)$$

donde  $V = num\_slots$

Una vez presentado el modelo, podemos generar la *matriz A*. Del mismo modo que pasaba en el modelo 1 tenemos que decidir qué criterios seguimos para definir organizar de forma secuencial cada uno de los estados, pues dependiendo de estos podremos identificar a qué estado corresponde cada una de las probabilidades que obtenemos.

Los criterios que vamos a aplicar son: en primer lugar en la lista de estados, figurarán los estados que tengan los slots reservados ocupados y los demás libres (siguiendo el orden de identificador de slot de mayor a menor para el tráfico de voz) y, en segundo lugar, aplicar este mismo orden del primer criterio pero con los datos ocupando uno de los slots compartidos (ocupando los slots con identificador de valor menor a mayor para el tráfico de datos y de mayor a menor para el tráfico de datos) y así hasta tener todas las combinaciones cubiertas. Siguiendo con el ejemplo anterior nos quedan los estados ordenados de la siguiente forma:

**Tabla 5.1** – Estados posibles con dos slots compartidos y dos reservados

Estado	Combinación			
0	0	0	0	0
1	V	0	0	0
2	0	V	0	0
3	V	V	0	0
4	0	0	0	D
5	V	0	0	D
6	0	V	0	D
7	V	V	0	D
8	0	0	D	0
9	V	0	D	0
10	0	V	D	0
11	V	V	D	0
12	0	0	D	D
13	V	0	D	D
14	0	V	D	D
15	V	V	D	D
16	V	V	V	0
17	V	V	0	V
18	V	V	V	D
19	V	V	D	V
20	V	V	V	V

De hecho este criterio se basa en la expresión 5.3 presentada anteriormente. Como tenemos dos slots de reserva para voz hay  $2^2$  (o sea cuatro) combinaciones posibles para estos slots. Es por esta razón que cada cuatro estados se repite la misma secuencia en los slots reservados para voz. En los compartidos vamos haciendo llegar datos. Para los estados del 16 al 20 es cuando introducimos las llamadas de voz en los slots compartidos, ya que no tenemos que olvidar que solo habrá voz en los slots compartidos cuando los reservados estén llenos.

De la misma forma que en el modelo uno (capítulo 4), obtenemos las ecuaciones de transición para todos los estados. A partir de aquí construiremos la matriz de transición de estados  $A$ .

Recordemos que cada fila indica los valores dependientes de cada ecuación de estado y la última fila es todo unos ya que substituye todos los valores del estado veinte para nuestro ejemplo, con la finalidad de provocar que la suma de probabilidades sea la unidad (condición indispensable). Introducimos otra diferencia respecto al primer modelo teniendo en cuenta que ahora podemos definir tres probabilidades de bloqueo, decidimos fijar las dos de voz (las de los slots de reserva y la total). Es por esta razón que substituímos una fila por unos, en los estados en los que se bloquean los slots de voz reservados, con la finalidad de forzar que la suma de estar en estas probabilidades sea la

probabilidad de bloqueo deseada. Estos estados son, si seguimos con el ejemplo son:

**Tabla 5.2** – Estados en los que están llenos los slots reservados

Estado	Combinación			
3	V	V	0	0
7	V	V	0	D
11	V	V	D	0
15	V	V	D	D
16	V	V	V	0
17	V	V	0	V
18	V	V	V	D
19	V	V	D	V
20	V	V	V	V

Las demás posiciones de la fila en cuestión las ponemos a 0.

De la misma forma, para forzar una  $PB_{total}$  de las llamadas de voz sustituimos en otra fila por uno en la posición correspondiente a los estados en que se produciría un bloqueo de las llamadas de voz. Estos estados, siguiendo con el ejemplo, son:

**Tabla 5.3** – Estados en que están llenos todos los slots

Estado	Combinación			
15	V	V	D	D
18	V	V	V	D
19	V	V	D	V
20	V	V	V	V

Las demás posiciones de la fila en cuestión las ponemos a 0. De esta forma obtenemos la matriz A.

También definimos una matriz  $P$  (siguiendo con el ejemplo de veintinueve elementos) tal que:

$$P = (P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}, P_{11}, P_{12}, P_{13}, P_{14}, P_{15}, P_{16}, P_{17}, P_{18}, P_{19}, P_{20})^T \tag{5.7}$$

y definimos en este caso la matriz de valores independientes  $B$  como:

$$B = (0 \ 0 \ 0 \ 0 \ 0 \ \dots \ 0 \ PB_{voz} \ PB_{res\_voz} \ 1)^T \tag{5.8}$$

con la finalidad de forzar que la suma de todas las probabilidades (recordemos que, para este caso, hemos substituido la última fila por unos) sea la unidad, que la suma de las probabilidades que hemos forzado en la penúltima fila sea la  $PB_{res\_voz}$  y que las de la antepenúltima fila sean la  $PB_{voz}$ . Así la ecuación del sistema es  $A \cdot P = B$  que al resolverlo, obtenemos las probabilidades de estar en cada estado.

## 5.2. Implementación del modelo

Empezamos la programación de este modelo creando un nuevo *script* en Matlab, pues no teníamos ninguna referencia. Un primer objetivo es que, dependiendo del número de slots total, el número de slots de voz y la probabilidad de bloqueo deseada, se puedan calcular las probabilidades de estar en cada estado. Consideramos un valor fijo de las tasa de llegadas de voz ( $\lambda_v$ ), la tasa de llegadas de datos ( $\lambda_d$ ). A partir de  $\lambda_v$  obtenemos, en primer lugar, con la ayuda de la PB y aplicando *Erlang B*, obtenemos el tiempo medio de llegadas ( $1/\mu_v$ ) correspondiente (ya hemos dicho que los slots reservados se comportan como un sistema de pérdidas puro) y, en segundo lugar, usando la fórmula mostrada anteriormente  $\lambda_{v,p}$ . Los pasos siguientes son los de crear la matriz *A*, generar la *B* y con todo esto calcular las probabilidades de los estados.

Anteriormente ya hemos comentado la importancia de saber en qué orden están las probabilidades, puesto que la posición de éstas en la matriz *A* afecta a la correcta interpretación de los datos finales. Para conseguir esto, programamos dentro el *script* una serie de instrucciones que, siguiendo los criterios expresados anteriormente, obtienen como resultado la matriz llamada *mloc*, que servirá de guía para crear la matriz *A* de forma correcta. Esta matriz contiene, dependiendo de los números de slots totales y el número de slots reservados para voz que desea el usuario, los estados ordenados según los criterios preestablecidos. Para poder ilustrarlo adecuadamente usamos el ejemplo anterior con dos slots reservados para voz y dos compartidos.

Los pasos seguidos por el *script* son:

En primer lugar realiza todas las combinaciones posibles con datos en uno de los slots compartidos. Tiene en cuenta que los datos llegan por el lado "derecho". Guarda las combinaciones en una matriz *auxiliar\_1*. Ejemplificado sería:

	O	O
	O	D
	D	O

Seguidamente coge esta matriz *auxiliar\_1* y para cada fila inserta en las columnas donde hay un 0 una D (simulando la llegada de datos). Para cada inserción antes de añadirla al final de la matriz *auxiliar\_1* comprueba que no exista ya en la matriz. Por ejemplo, si coge la primera fila de la matriz, que es 0 0 e introduce una D quedará '0 D'. Sin embargo esta combinación no la pondrá

en la matriz *auxiliar\_1* pues al mirar si ya está introducida verá que sí lo está en la segunda fila. Así, veremos que solo añade una fila:

	O	O
	O	D
	D	O
	D	D

De la misma forma realiza todas las combinaciones posibles con voz en uno de los slots reservados. Teniendo en cuenta que las llamadas de voz llegan por el lado "izquierdo", guarda las combinaciones en otra matriz *auxiliar\_2*. Gráficamente sería:

	O	O
	V	O
	O	V

De la misma forma que se ha hecho con los datos, coge la matriz *auxiliar\_2* y para cada fila inserta en las columnas donde haya un 0 una V (simulando la llegada de voz). Antes de insertar la nueva combinación en la matriz *auxiliar\_2* comprueba que la combinación no exista ya en la *auxiliar\_2*. El resultado (teniendo en cuenta el ejemplo):

	O	O
	V	O
	O	V
	V	V

Seguidamente cruza los datos de las dos matrices auxiliares. Lo hace con dos bucles, uno dentro del otro, que recorren las dos matrices de forma que coge la primera fila de la *auxiliar\_1* y la une a cada una de las filas de *auxiliar\_2*. El resultado se guarda en la matriz *miloc*. La primera iteración sería:

	O	O	O	O
	O	O	O	D
	O	O	D	O
	O	O	D	D

El bucle continua hasta recorrer todas las filas de la matriz *auxiliar\_1*. En el ejemplo obtendríamos:

	O	O	O	O
	O	O	O	D
	O	O	D	O
	O	O	D	D
	V	O	O	O
	V	O	O	D
	V	O	D	O

	V	O	D	D
	O	V	O	O
	O	V	O	D
	O	V	D	O
	O	V	D	D
	V	V	O	O
	V	V	O	D
	V	V	D	O
	V	V	D	D

Como vemos aún falta tener en cuenta los estados en que voz y datos conviven en los slots compartidos. Es por eso que coge la matriz *auxiliar\_1* y, para cada fila, inserta en las columnas donde haya un 0 una V (simulando la llegada de voz). El resultado lo guarda en una matriz *auxiliar\_3*, Pero antes de hacerlo comprueba que la combinación no exista ya en la *auxiliar\_3*. El resultado es:

	V	O
	O	V
	V	D
	D	V
	V	V

Una vez llegados a este punto, como sabemos que estos casos solo se dan si los slots reservados están llenos (mirar modelo) acaba de llenar la matriz *mloc*, añadiendo tantas filas como filas tiene *auxiliar\_3* y poniendo para cada fila en las columnas correspondientes a los slots reservados V y a los compartidos lo que contiene *auxiliar\_3*. De esta forma *mloc* queda de manera definitiva como:

	O	O	O	O
	O	O	O	D
	O	O	D	O
	O	O	D	D
	V	O	O	O
	V	O	O	D
	V	O	D	O
	V	O	D	D
	O	V	O	O
	O	V	O	D
	O	V	D	O
	O	V	D	D
	V	V	O	O
	V	V	O	D
	V	V	D	O
	V	V	D	D
	V	V	V	O
	V	V	O	V
	V	V	V	D
	V	V	D	V



Una vez tenemos generada la matriz *mloc* ya puede empezar a generar la matriz *A*. En primer lugar, la crea y la inicializa toda a 0, después crea un bucle (con la auxiliar *i*) que recorra toda la matriz *mloc*. Ya sabemos que dimensión tiene esta matriz, pues conocemos el número de slots totales y el número de estados que hay para estas condiciones (ver fórmula 5.3). Para cada fila de *mloc* (a las que apuntamos, recordemos, con *i*) mira cuantas *V* hay en los slots de reserva (llamamos a la auxiliar *ves*), cuantas *V* en los compartidos (*ves2*) y cuantas *D* en los compartidos (*des*). Con la finalidad de poder determinar desde qué estados podemos venir y a qué estados podemos ir (el que apunta el primer bucle). Dentro del primer bucle empieza otro bucle (con la auxiliar *j*) que también recorre todas las filas de *mloc*, y mira cuantas *V* hay en los slots de reserva (*vesj*), cuantas *V* en los compartidos (*vesj2*) y cuantas *D* en los compartidos (*desj*). En este momento y, dependiendo de la relación existente entre las auxiliares, va creando la matriz *A*:

- Si  $des == desj$  y  $(ves + ves2 + 1) == vesj2 + vesj$   
 y  $[ves == vesj$  y  $vesj2 == ves2 + 1]$  o  $[vesj2 == 0$  y  $vesj - 1 == ves]$

y las dos filas que estamos estudiando son iguales en todos sus elementos menos en uno => en la columna de la fila *i* correspondiente a *j* inserta un  $-\mu_v$ .

o Ejemplo

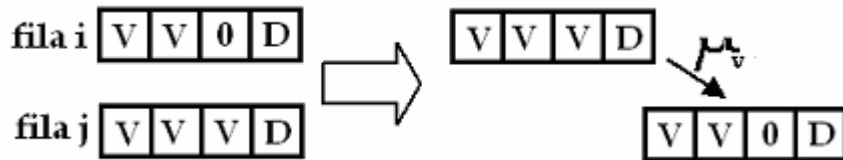


Fig 5.11

pues :  $ves=2$        $vesj=2$   
           $ves2=1$      $vesj2=0$   
           $des=1$        $desj=1$

- Si:  $ves+ves2 == vesj+vesj2$  y  $des == desj - 1$  ] y las dos filas que estamos estudiando son iguales en todos sus elementos menos en uno => en la columna de la fila *i* correspondiente a *j* inserta un  $-\mu_v$ .

▪ Ejemplo

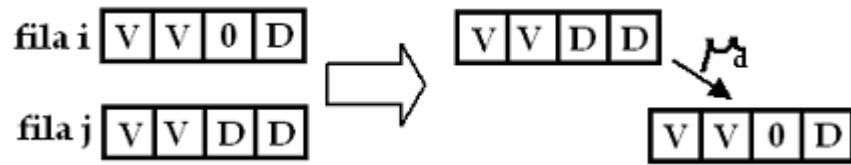


Fig. 5.12

pues:  $ves=2$        $vesj=2$   
           $ves2=0$       $vesj2=0$   
           $des=1$          $desj=2$

- Si :  $ves+ves2==vesj+vesj2$  y  $des-1==desj$  y las dos filas que estamos estudiando son iguales en todos sus elementos menos en uno => en la columna de la fila i correspondiente a j inserta un  $-?_d$ .

- Ejemplo

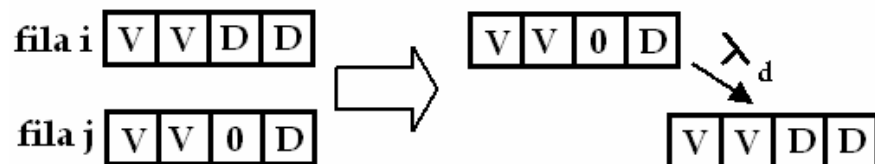


Fig. 5.13

pues:  $ves=2$        $vesj=2$   
           $ves2=0$       $vesj2=0$   
           $des=2$          $desj=1$

- Si :  $des==desj$  y  $ves-1==vesj$   $desj$  y las dos filas que estamos estudiando son iguales en todos sus elementos menos en uno => en la columna de la fila i correspondiente a j inserta un  $-?_v$ .

- Ejemplo

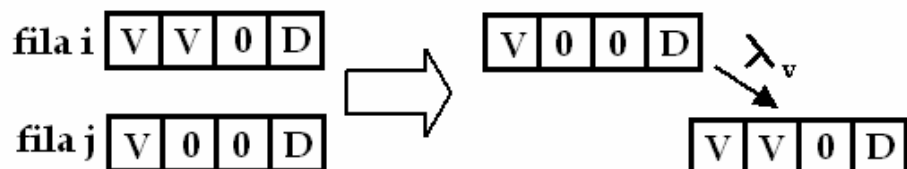


Fig 5.14

pues:  $ves=2$        $vesj=1$



ves2=0      vesj2=0  
des=1      desj=1

Si:  $des==desj$  y  $ves2-1==vesj2$  y  $ves==vesj$  y  $ves==num\_slots\_voz$  y las dos filas que estamos estudiando son iguales en todos sus elementos menos en uno  $\Rightarrow$  en la columna de la fila  $i$  correspondiente a  $j$  inserta un  $\lambda_{v_r}$

- Ejemplo
- 

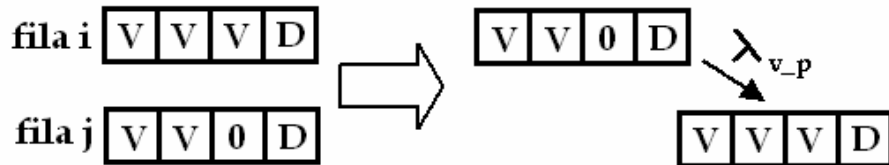


Fig. 5.15

pues: ves=2      vesj=2  
ves2=1      vesj2=0  
des=1      desj=1

Seguidamente crea la diagonal. Lo que hace es mirar la trama slot por slot. Sabemos que para cada D o V que haya tendrá que poner una  $\mu_d$  o  $\mu_v$  respectivamente en la diagonal de la fila correspondiente. En el caso de este ejemplo:



Fig. 5.16

Recorre la fila de izquierda a derecha. En el primer slot encuentra una V e inserta una  $\mu_v$  en la posición  $(i,i)$  de la matriz A. Mira el segundo y ve que hay otra V, cosa que hace que inserte una  $\mu_v$  en la posición  $(i,i)$  de la matriz A que se suma a lo que ya había anteriormente. En el tercer slot no encuentra ni una V ni una D mientras que en el cuarto encuentra una D que provoca que en la posición  $(i,i)$  de la matriz A se sume una  $\mu_d$  a lo que ya había. En este momento la fila  $i$  en la columna  $i$  tendría la forma:

$$\mu_v + \mu_v + \mu_d$$

Para las lambdas el *script* lo que hace es que, para cada fila, recorrer en primer lugar los slots compartidos de derecha a izquierda y mirar donde esta el primer '0'. El valor de la posición lo guarda en una variable llamada *lastzerod*. El hecho de que esta variable tenga un valor mayor que cero ya asegura que se puede ir con  $\lambda_d$  desde el estado presente a otro, con lo que tenemos que introducir una  $\lambda_d$  en la diagonal. Para las v el *script* recorre la fila de izquierda a

derecha y mira donde esta el primer '0'. El valor de la posición lo guarda en una variable llamada *lastzerov*. Pueden pasar dos cosas:

- Si *lastzerov* es menor que el número de slots reservados quiere decir que se puede ir con  $?_v$  des del estado presente a otro, pues los slots reservados no están llenos.
- Si *lastzerov* es mayor que el número de slots reservados quiere decir que para ir del estado presente al otro con la llegada de una llamada de voz se debe hacer con  $?_{v_r}$ , pues los slots reservados están llenos y la llamada se desborda de estos y va hacia los compartidos.

Si miramos el ejemplo veremos que en primer recorrido de derecha a izquierda ve un 0 en los slots compartidos que hace aumentar la variable *lastzerod*, por lo que, al tomar *lastzerod* un valor mayor que uno inserta una  $?_d$  en la posición (i,i) de la matriz A que se suma a lo que ya había anteriormente:

$$\mu_v + \mu_v + \mu_d + ?_d$$

Seguidamente recorre la fila de izquierda a derecha y ve que el primer 0 está en la posición 3. Este valor lo guarda en *lastzerov*, y ve que es mayor que el número de slots de voz reservados (recordemos que son dos). Por lo tanto inserta una  $?_{v_r}$  en la posición (i,i) de la matriz A que se suma a lo que ya había anteriormente. Así, el valor A(i,i) queda:

$$\mu_v + \mu_v + \mu_d + ?_d + ?_{v_r}$$

## 6. ANÁLISIS DE PROBABILIDADES DE ENCONTRAR RÁFAGAS

### 6.1. Implementación

Una vez tenemos claro cómo calcular las probabilidades de estar en cada estado, lo que realmente es interesante para nuestro estudio es saber las probabilidades que hay de que encontremos ráfagas de un determinado número de slots consecutivos vacíos. Esto es lo que realmente nos sirve para verificar si un modelo es más eficiente que otro, con esta finalidad creamos un *script*.

Los pasos que realiza el *script* son: en primer lugar crea una matriz, que llamamos *m\_rafagas*, que contiene, para cada estado posible, la posición en que empieza una ráfaga, es decir, donde aparece el primer 0, y cuantos 0 más hay. El número de filas de *m\_rafagas* será siempre igual al número de estados que estamos tratando en nuestro modelo. Por ejemplo, para '0 0 1 0 0 1 1' la fila de *m\_rafagas* correspondiente a este estado tendría el valor 2 0 0 2 0 0 0, que indicaría que en la posición 1 empieza una ráfaga de dos ceros y que en la posición cuatro empieza otra ráfaga de dos ceros. Esta metodología es la que permite crear la matriz *m\_rafagas* para todos los estados posibles.

Una vez obtiene la tiene matriz *m\_rafagas*, a partir de los valores de las probabilidades de estar en cada uno de los estados, vector P, el procedimiento con los contenidos en P, que recordemos que son las probabilidades de estar en cada uno de los estados. El procedimiento es: en primer lugar crea un vector *v\_probs\_rafagas* con tantas posiciones como ráfagas de slots posibles (por ejemplo en el caso de que tengamos tres slots podrá haber ráfagas de un, dos o tres slots, por lo tanto el vector tendrá tres posiciones) que acumula para cada posición las probabilidades de que haya una ráfaga de cada una de las posibles longitudes. *V\_probs\_rafagas* se va completando en función de cada fila de *m\_rafagas*. Cuando encuentra, en *m\_rafagas*, un elemento distinto de cero se acumula el valor de la probabilidad en el vector *v\_probs\_rafagas*. La posición que se actualiza de *v\_probs\_rafagas* es justo el valor distinto de 0 que encontramos en *m\_rafagas*.

Por ejemplo, supongamos 7 slots:

Vector P:	P(estar en el estado N)= 0,2
Matriz <i>m_rafagas</i> :	Fila N ---> 2 0 0 0 0 0 0
<i>V_probs_rafagas</i> :	Valor Inicial ---> 0 0 0 0 0 0 0

Recorre el primer elemento de la fila N de *m\_rafagas* y hay un 2. Esto significa que en la posición 2 del vector *v\_probs\_rafagas* pondremos el valor

correspondiente a la probabilidad de estar en el estado N (fila N). De esta forma para este ejemplo  $v\_probs\_rafagas$  quedaría '0 0,2 0 0 0 0 0'

## 7. RESULTADOS

Siguiendo el modelo 2 (ver capítulo 5), hemos realizado los cálculos para los casos en los que se reserven 2, 3 y 4 slots para las llamadas de voz.

Suponemos fijos los parámetros correspondientes al tráfico de datos. Así,  $\lambda_d = 2$  paquetes/segundo y  $\mu_d = 2$  paquetes/segundo. Los valores de voz se escogen basándose en la probabilidad de bloqueo de los slots reservados exclusivamente para voz ( $PB_{res\_voz}$ ). Miramos en la tabla de Erlang para el número de slots reservados que tenemos y la probabilidad deseada que valor de A le corresponde. Una vez tenemos el valor del tráfico (A) y conociendo que en el caso de llamadas de voz los valores que toma  $1/\mu_v$  van de 60 a 180 segundos obtenemos el valor de  $\lambda_{v\_r}$ . Por ejemplo si suponemos que tenemos 3 slots de reserva y queremos un  $PB_{res\_voz}$  de 5%, vemos en la tabla de Erlang (imagen 7.1) que le corresponde una A de 0.9. Suponiendo una  $1/\mu_v$  de 60 obtenemos que  $\lambda_{v\_r}$  es de 0.015.

A \ C	3
0.7	028552
0.8	038694
0.9	050072

**Fig. 7.1** – Detalle de la tabla de Erlang-B

Usando la fórmula 5.1 obtenemos  $\lambda_v$  que es de 0.0045. Introducimos todos estos valores en el *script* de Matlab. Mediante la variación de los valores de los coeficientes independientes del sistema de ecuaciones original, B (ver Modelo 2) del valor correspondiente a la  $PB_{total}$  obtenemos las probabilidades de estar en cada estado (las consideramos correctas para el valor de  $PB_{total}$  en el que son todas positivas y cumplen las condiciones marcadas).

A destacar que no obtenemos siempre para los valores introducidos un único valor de  $PB_{total}$  que haga cumplir los requisitos que queremos, sino que es posible que dentro de un abanico de valores de  $PB_{total}$  obtengamos resultados coherentes. Esto pasa ya que el modelo en el que comparten slots voz y datos no es un sistema de pérdidas puro, y no intervienen solo las *lambdas* y las *mus* respectivas sino también valores medios y variaciones que provocan esta circunstancia.

## 7.1. General

### 7.1.1 Caso 1

$$PB_{\text{total}} = 3\% \quad N^{\circ} \text{ slots reserva} = 4 \quad N^{\circ} \text{ slots compartidos} = 3$$

**Tabla 7.1** – Probabilidad de ráfagas de 1,2 o 3 slots para distintas  $\mu_v$ .

$\mu_v$	Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots
1/60	0,3632	0,28159	0,35521
1/80	0,36329	0,2816	0,3551
1/100	0,36335	0,28161	0,35503
1/120	0,36339	0,28162	0,35499
1/140	0,36342	0,28162	0,35496
1/160	0,36344	0,28162	0,35494
1/180	0,36346	0,28163	0,35492

Podemos observar que con los valores de  $\mu_v$  en que nos movemos, las probabilidades de estar en cada estado no cambian de manera significativa. Por ejemplo, para la ráfaga de un slot con una  $\mu_v$  de 1/60 la probabilidad de tener dicha ráfaga es del 36,32% mientras que con  $\mu_v$  de 1/180 es del 36,34%, una diferencia de 0.02% que consideramos despreciable. Es por esta razón que los siguientes casos que presentamos tienen  $\mu_v = 1/60$ .

## 7.2. Número de slots fijo vs. $PB_{\text{total}}$ variable

### 7.2.1. Caso 1

Cálculo 1:

$$PB_{\text{total}} = 1\% \quad N^{\circ} \text{ slots reserva} = 2 \quad N^{\circ} \text{ slots compartidos} = 5$$

**Tabla 7.2** – Probabilidad de ráfagas de 1,2,3,4 y 5 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
5,4767 %	12,39 %	21,195 %	26,263 %	34,676 %

Cálculo 2:

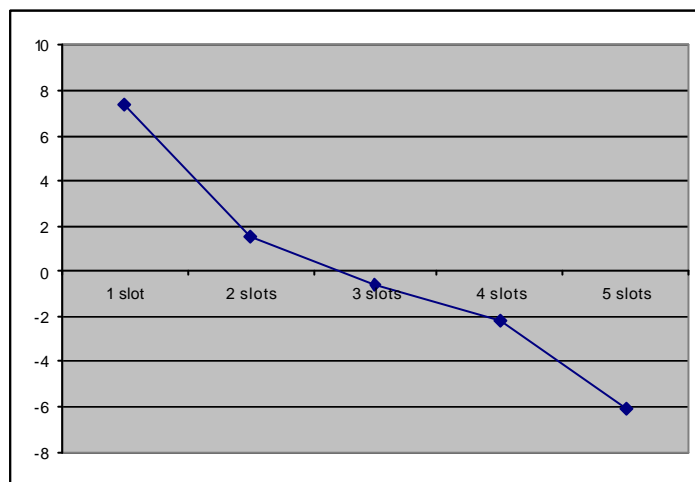
$PB_{total} = 4\%$        $N^{\circ}$  slots reserva = 2       $N^{\circ}$  slots compartidos = 5

**Tabla 7.3** – Probabilidad de ráfagas de 1,2,3,4 y 5 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
12,86 %	13,947 %	20,538 %	24,061 %	28,594 %

**Tabla 7.4** – Diferencia (Cálculo 2 – Cálculo 1 1)

	Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
Diferencia (Cálculo 2 – Cálculo 1)	7,38 %	1,557 %	-0,657 %	-2,201 %	-6.082 %



**Fig. 7.2**

### 7.2.2. Caso 2

Cálculo 1:

$PB_{total} = 1,5\%$        $N^{\circ}$  slots reserva = 3       $N^{\circ}$  slots compartidos = 4

**Tabla 7.5** – Probabilidad de ráfagas de 1, 2,3 y 4 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots
14,221	23,607	26,744	35,427

Cálculo 2:

$PB_{total} = 4\%$

Nº slots reserva = 3

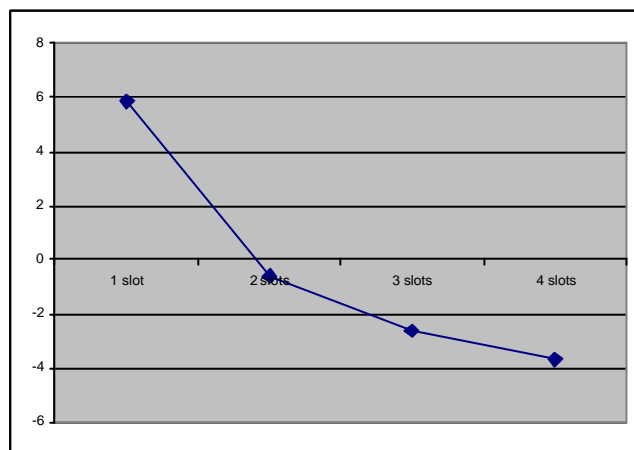
Nº slots compartidos = 4

**Tabla 7.6** – Probabilidad de ráfagas de 1,2,3 y 4 slots para  $1/\mu_V=60$ .

Ráfaga 1 sbt	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots
20,085	23,007	25,15	31,759

**Tabla 7.7** – Diferencia (Cálculo 2 – Cálculo 1)

	Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots
Diferencia (Cálculo 2 – Cálculo 1)	5,864 %	-0,6 %	-2,624 %	-3,668%



**Fig. 7.3**

### 7.2.3. Caso 3

Cálculo 1:

$PB_{total} = 0,8\%$

Nº slots reserva = 4

Nº slots compartidos = 3



**Tabla 7.8** – Probabilidad de ráfagas de 1,2 y 3 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots
34,044	29,137	36,819

Cálculo 2:

 $PB_{total} = 5,8\%$ 

Nº slots reserva = 4

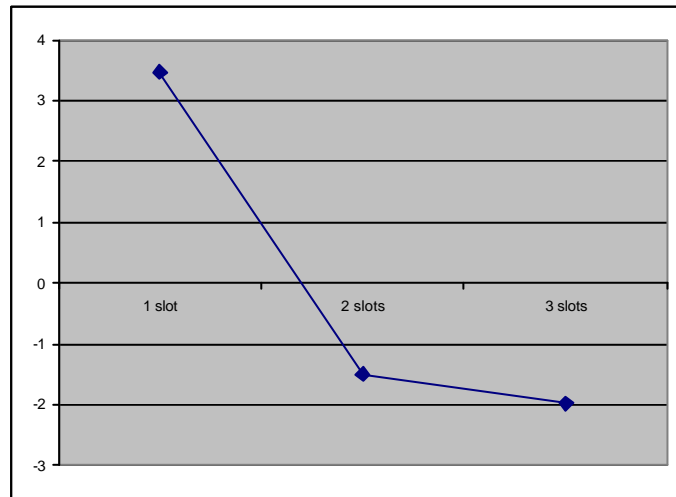
Nº slots compartidos = 3

**Tabla 7.9** – Probabilidad de ráfagas de 1,2 y 3 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots
37,522	27,637	34,841

**Tabla 7.10** – Diferencia (Cálculo 2 – Cálculo 1)

	Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots
Diferencia (Cálculo 2 – Cálculo 1)	3,478%	-1,5 %	-1,978 %

**Fig. 7.4**

#### 7.2.4. Comentario

En el caso que comparemos dos simulaciones con igual número de slots y distinta  $PB_{total}$  vemos que lo que ocurre es que como más alta es la  $PB_{total}$  más

aumenta la probabilidad de tener ráfagas de menos slots y disminuye la de tener ráfagas de más slots. Concretamente, para el caso 1 si aumenta la  $PB_{total}$  la probabilidad de tener ráfagas de 1 y 2 slots aumenta y la de 3, 4 y 5 slots disminuye respecto al cálculo con menor  $PB_{total}$ . Para los casos 2 y 3 aumenta la probabilidad de tener ráfagas de 1 slot y disminuye la de tener ráfagas de 2 y 3. En particular en el caso 2 la tener probabilidad de tener ráfagas de 4 slots también aumenta si disminuimos la  $PB_{total}$ .

Conclusión – Cuando contrastamos los dos casos con igual número de slots de reserva y distinta  $PB_{total}$ , conseguimos probabilidades mayores de tener ráfagas con más slots (ráfagas largas) si fijamos la  $PB_{total}$  con un valor pequeño.

### 7.3. Número de slots variable vs. $PB_{total}$ fijo

#### 7.3.1 Caso 1

Cálculo 1:

$PB_{total} = 1,5\%$

Nº slots reserva = 2

Nº slots compartidos = 5

**Tabla 7.11** – Probabilidad de ráfagas de 1,2,3,4 y 5 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
6,9933 %	12,772 %	21,092 %	25,874 %	33,269 %

Cálculo 2:

$PB_{total} = 1,5\%$

Nº slots reserva = 3

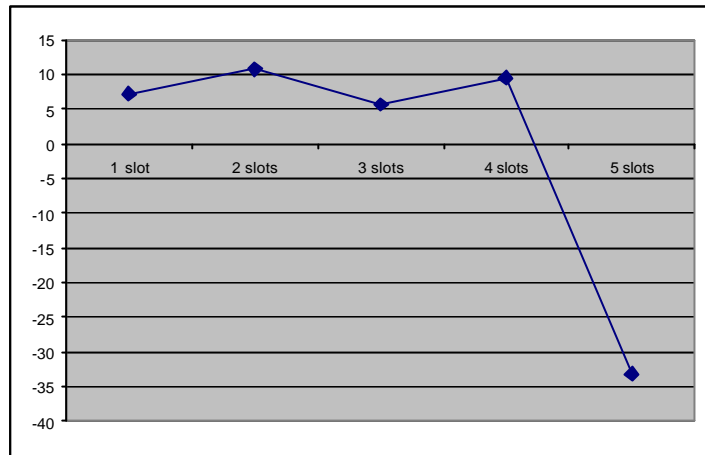
Nº slots compartidos = 4

**Tabla 7.12** – Probabilidad de ráfagas de 1,2,3 y 4 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots
14,221 %	23,607 %	26,744 %	35,427 %

**Tabla 7.13** – Diferencia (Cálculo 2 – Cálculo 1)

	Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
Diferencia (Cálculo 2 – Cálculo 1)	7.2 %	10.8 %	5.652 %	9.5 %	-33.2 %

**Fig. 7.5****7.3.2 Caso 2**

Cálculo 1:

 $PB_{total} = 1,5\%$ 

Nº slots reserva = 2

Nº slots compartidos = 5

**Tabla 7.14** – Probabilidad de ráfagas de 1,2,3,4 y 5 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
6,9933 %	12,772 %	21,092 %	25,874 %	33,269 %

Cálculo 2:

 $PB_{total} = 1,5\%$ 

Nº slots reserva = 4

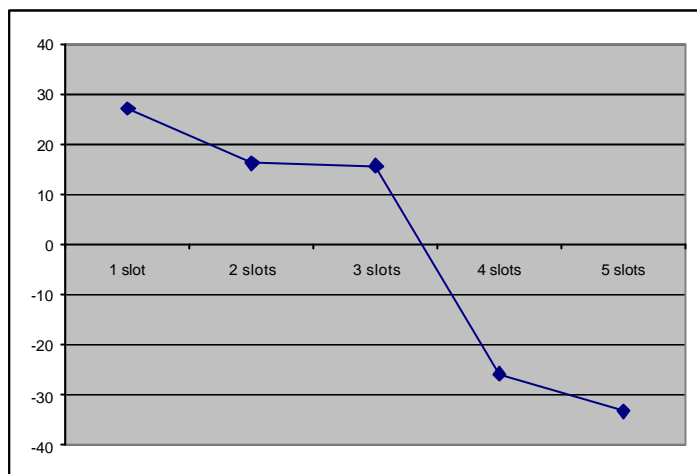
Nº slots compartidos = 3

**Tabla 7.15** – Probabilidad de ráfagas de 1,2 y 3 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots
34,157 %	29,061 %	36,782 %

**Tabla 7.16** – Diferencia (Cálculo 2 – Cálculo 1)

	Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
Diferencia (Cálculo 2 – Cálculo 1)	27,1 %	16,2 %	15,69 %	-25,8 %	-33,2 %

**Fig. 7.6****7.3.3. Caso 3**

Cálculo 1:

 $PB_{total} = 1,5\%$ 

Nº slots reserva = 3

Nº slots compartidos = 4

**Tabla 7.16** – Probabilidad de ráfagas de 1,2,3 y 4 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots
14,221 %	23,607 %	26,744 %	35,427 %

Cálculo 2:

 $PB_{total} = 1,5\%$ 

Nº slots reserva = 4

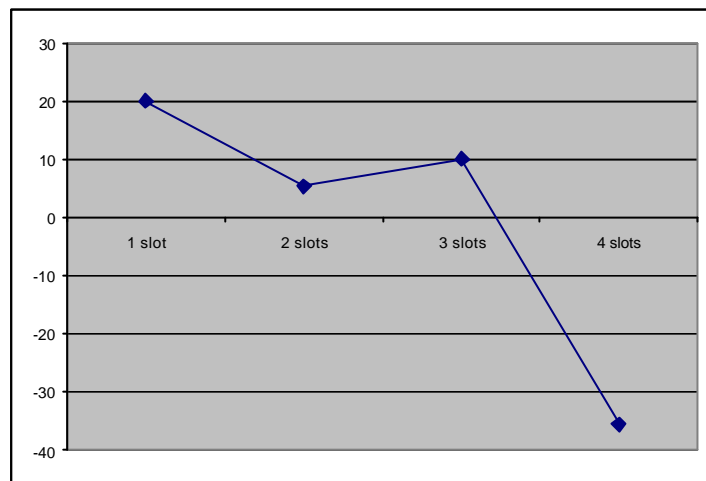
Nº slots compartidos = 3

**Tabla 7.17** – Probabilidad de ráfagas de 1,2 y 3 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots
34,157 %	29,061 %	36,782 %

**Tabla 7.18** – Diferencia (Cálculo 2 – Cálculo 1)

	Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots
Diferencia (Cálculo 2 – Cálculo 1)	19,93 %	5,45 %	10,038 %	-35,4

**Fig. 7.7**

### 7.3.4. Comentario

Para analizar este contexto hemos de tener en cuenta las longitudes de las ráfagas con las que queremos trabajar. Esta aclaración es importante puesto que sería fácil pensar que sería de nuestro interés el menor número de slots compartidos (en nuestro caso 2 slots compartidos) para poder obtener ráfagas de 5 slots que con los otros cálculos propuestos serían inviables.

Intentando encontrar el número de slots de reserva con los que obtenemos mejores probabilidades de ráfagas, podemos observar en las gráficas anteriores que para cada caso estudiado: no obtenemos mejores resultados (con la misma  $PB_{total}$ ) para ningún caso en concreto, ya que depende de la longitud de ráfaga con la que deseamos trabajar, la elección del número de slots de reserva y de slots compartidos que más nos convenga.

Por ejemplo, si queremos ráfagas de 3 slots obtendremos con 2 slots de reserva una probabilidad del 21,1%, con 3 de reserva un 26,7% y con 4 un 36,7%. Pero esto no es del todo cierto, puesto que una ráfaga de 4 slots incluye una ráfaga de 3 slots, de la misma forma que una de 5 incluye una de 4, una de 3 y dos de 2.

De esta forma para todos los casos expuestos, si acumulamos probabilidades será más favorable cuanto menos slots utilizemos (trabajando en todos los casos con la misma  $PB_{total}$ ). De esta forma, y para seguir con el ejemplo anterior de una ráfaga de 3 slots, con 2 slots de reserva la probabilidad es de 80%, con 3 de reserva de 62,1% y con 4 de 36,7%.

Conclusión - > En el caso que con la misma  $PB_{total}$  comparemos entre un número de slots de reserva u otro, deberemos escoger el menor número de slots, consecuentemente obtendremos probabilidades de tener ráfagas mayores que se podrán acumular para obtener la ráfaga de longitud deseada.

## 7.4. Número de slots variable vs. $PB_{total}$ variable

### 7.4.1. Caso 1

Cálculo 1:

$PB_{total} = 0,01\%$                        $N^{\circ}$  slots reserva = 2                       $N^{\circ}$  slots compartidos = 5

**Tabla 7.19** – Probabilidad de ráfagas de 1,2,3,4 y 5 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
3,5167 %	11,864 %	21,272 %	26,548 %	36,799 %

Cálculo 2:

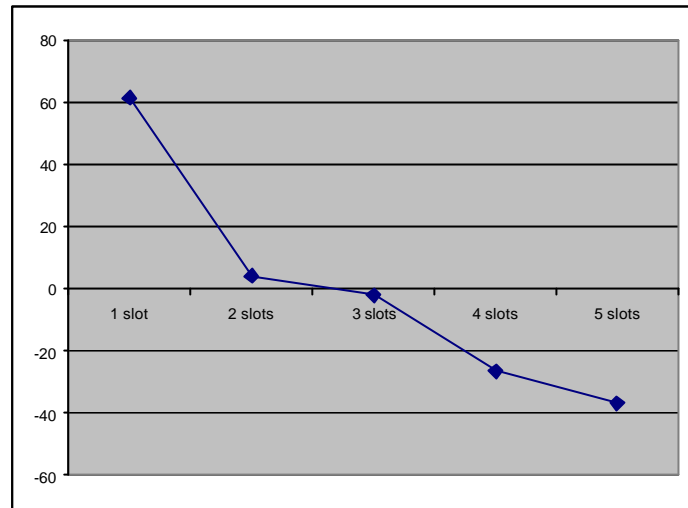
$PB_{total} = 41\%$                        $N^{\circ}$  slots reserva = 4                       $N^{\circ}$  slots compartidos = 3

**Tabla 7.20** – Probabilidad de ráfagas de 1,2 y 3 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots
64,827 %	15,918 %	19,255 %

**Tabla 7.21** – Diferencia (Cálculo 2 – Cálculo 1)

	Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
Diferencia (Cálculo 2 – Cálculo 1)	61,3 %	4,036 %	-2,017 %	-26,548 %	-36,799 %

**Fig. 7.8****7.4.2. Caso 2**

Cálculo 1:

 $PB_{total} = 30\%$ 

Nº slots reserva = 2

Nº slots compartidos = 5

**Tabla 7.22** – Probabilidad de ráfagas de 1,2,3,4 y 5 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
51,201 %	12,813 %	12,272 %	11,783 %	11,932 %

Cálculo 2:

 $PB_{total} = 0,8\%$ 

Nº slots reserva = 4

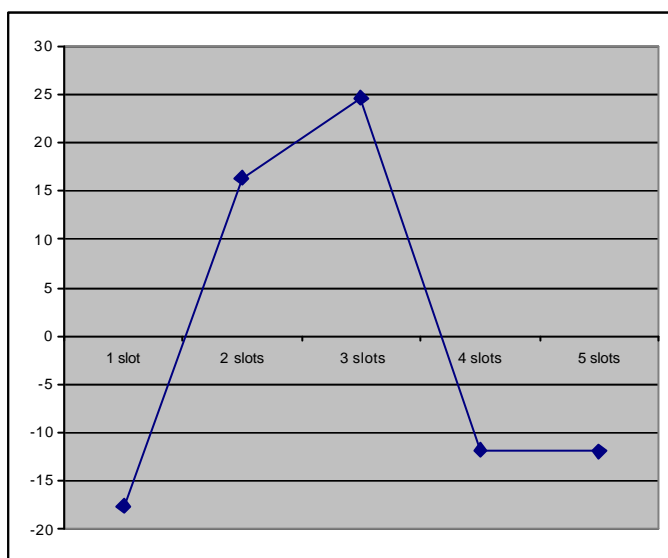
Nº slots compartidos = 3

**Tabla 7.23** – Probabilidad de ráfagas de 1,2 y 3 slots para  $1/\mu_v=60$ .

Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots
34,044 %	29,137 %	36,819 %

**Tabla 7.24** – Diferencia (Cálculo 2 – Cálculo 1)

	Ráfaga 1 slot	Ráfaga 2 slots	Ráfaga 3 slots	Ráfaga 4 slots	Ráfaga 5 slots
Diferencia (Cálculo 2 – Cálculo 1)	-17,57 %	16,324 %	24,547 %	-11,783 %	-11,932 %

**Fig. 7.9**

### 7.4.3. Comentario

Tal como se muestra en las figuras anteriores, podemos observar que el cálculo con una  $PB_{total}$  alta independientemente de los slots de reserva que tenga, implica que la probabilidad de encontrar una ráfaga con 1 slot es muy alta, mientras que una  $PB_{total}$  muy baja conlleva que la probabilidad de ráfagas de más slots aumente. No hay que caer en el error de decir que siempre hay más probabilidad de que haya una ráfaga de 4 o 5 slots independientemente de la  $PB_{total}$ . La razón es clara: estamos comparando dos situaciones, una con 5 slots compartidos y otra con 3, de tal forma que las probabilidades de tener ráfagas de 4 y 5 slots son exclusivas de los cálculos de 2 slots de reserva, pues en el otro caso no existe. En las gráficas anteriores también se pudo observar que cuanto más aumentemos la  $PB_{total}$ , tanto para el caso de 2 slots de reserva como para el de 4 slots de reserva, la tendencia es la de aumentar la probabilidad de tener ráfagas de 1 slot, mientras que las otras probabilidades



tienden a tener valores similares, en el caso de 4 slots de reserva alrededor del 15-20% y en el de 2 slots del 10%.

Conclusión- Se corrobora lo dicho en el análisis del caso 2: interesa una  $PB_{total}$  baja para conseguir una probabilidad de ráfagas de mayor longitud alta.

## 7.5 Conclusiones

Después de realizar los cálculos de los cuatro casos anteriores, podemos decir que:

- Si se aumenta la  $PB_{total}$  (independientemente del número de slots de reserva para voz)
  - Las probabilidades de tener ráfagas grandes disminuyen y aumentan las de tener ráfagas cortas (1 o 2 slots).
  - La PB de voz aumenta.
  - Por todo esto decimos que disminuye la QoS tanto para voz como para el tráfico de datos.
  
- El número de slots de reserva debe ser siempre el menor posible, así obtenemos ráfagas mayores que después se pueden separar para conseguir ráfagas más pequeñas.

Con las dos condiciones podemos decir que en general, nos interesa tener tanto la  $PB_{total}$  como el número de slots de reserva para voz lo más bajos que permita el sistema. Si observamos los resultados presentados anteriormente vemos que una disminución de la  $PB_{total}$  aumenta la probabilidad de tener la máxima ráfaga posible.

Si bien a simple vista parece que una  $PB_{total}$  menor no influya en gran medida en los resultados, es cierto que una menor  $PB_{total}$  va relacionada con la QoS tanto para voz como para datos que queramos proporcionar. Esto se debe a que las probabilidades de tener ráfagas de distinta longitud difieren en menos de un 8% siempre que la  $PB_{total}$  no sobrepase un 5% - límite para una QoS aceptable- (ver 7.1). El tráfico que va ocupando los slots compartidos procede (el de voz en concreto) del tráfico desbordado debido que los slots reservados están ocupados. Si la PB de los slots reservados es muy alta llegará más tráfico y aumentará la PB de los slots compartidos, empeorando la QoS tanto para voz como para datos, pues se perderán peticiones.

La QoS también se ve afectada por el número de slots de reserva para voz que determinemos. Cuanto menos slots de reserva para voz se disponga, más slots compartidos habrá, y por lo tanto susceptibles de ser ocupados por tráfico de datos. Esto puede ser peligroso pues se podría dar el caso que la mayoría de

los slots compartidos se llenasen de ráfagas de datos que hiciesen que muchas llamadas de voz no pudiesen ser atendidas, disminuyendo la QoS de los usuarios.

En definitiva, nos tenemos que basar en dos cosas:

- $PB_{total}$  la menor posible
- Número de slots de reserva para voz el menor posible siempre teniendo en cuenta la QoS (tanto de voz como de datos) que se quiera ofrecer al usuario.

## **8. ESTUDIO DE AMBIENTALIZACIÓN**

Sabemos que las comunicaciones en general y las comunicaciones móviles en particular disminuyen el uso de los medios de transporte contaminantes para el medio ambiente.

Las redes móviles celulares pueden tener un impacto medioambiental en cuanto a la ubicación de los subsistemas de estaciones móviles (BSS), no obstante nuestro trabajo no está centrado en la planificación de estas redes, sino más bien en una optimización interna de la gestión de los recursos radio, por tanto, no añade ningún tipo de impacto ambiental.

## 9. BIBLIOGRAFIA

- [1] [www.3gamericas.org](http://www.3gamericas.org)
- [2] [www.umtsforum.net](http://www.umtsforum.net)
- [3] Hamiti S. y otros, *GSM/EDGE Radio Access Network (GERAN) – Evolution of GSM/EDGE towards 3G Mobile Services*
- [4] Martínez J. y Casares V., *Servicios portadores en redes celulares 2,5G y 3G y su dimensionado*, Universidad Politécnica de Valencia
- [5] 3GPP TS 25.240, *UTRAN Iur Interface General Aspects and Principles*, Release 1999
- [6] 3GPP TS 43.051, *Digital cellular telecommunications system (Phase 2+); GSM/EDGE Radio Access Network (GERAN) overall description; version 6.0.0*, Release 6, March 2005
- [7] White Paper: EDGE Introduction of high-speed data in GSM/GPRS Networks, Ericsson AB, 2003
- [8] Itkonen J. y Lempiäinen, J., *Air Interface Capacity for GPRS/EDGE over GSM Traffic Load*, IEEE, Helsinki, 2002
- [9] Ivanov K., Ball C.F., y Trembl F., *GPRS / EDGE Performance on Reserved and Shared Packet Data Channels*, IEEE, Munich, 2003
- [10] Ni S. y Haggman S.-G., *GPRS performance estimation in GSM circuit switched services and GPRS shared resource systems*, pp. 1417-1421, IEEE, Helsinki, 1999
- [11] Fitzpatrick P., Ivanovich M. y Yin J., *Models for Pre-emption of Packet Data by Voice in Slotted Cellular Radio Networks*, pp. 1698-1702, IEEE, Australia, 2002
- [12] Ajib W., Godlewski P., *Effects of Circuit Switched Transmissions over GPRS Performance*, ACM, Paris, 2000
- [13] Chen W., Wu J-L., Liu H., *Performance Analysis of Radio Resource Allocation in GSM/GPRS Networks*, IEEE, Taipei, 2002
- [14] Ramírez C. y Paradells J., *Documento interno del Departamento de Ingeniería Telemática de la Universidad Politécnica de Catalunya*, Barcelona, 2005
- [15] [www.mathworks.com/access/helpdesk/help/techdoc/ref/dec2bin.html](http://www.mathworks.com/access/helpdesk/help/techdoc/ref/dec2bin.html)



**Escola Politècnica Superior  
de Castelldefels**

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# ANNEXOS

**TÍTOL DEL TFC: Software de gestión en redes móviles**

**TITULACIÓ: Enginyeria Tècnica de Telecomunicació, especialitat Telemàtica**

**AUTOR: Jordi Collell i Codina**

**DIRECTOR: Carlos M. Ramírez Casañas**

**DATA: 22 de febrer de 2006**



**ANEXOS: Parámetros de tráfico****General**

Nº slots reserva	$\lambda_{v_r}$	$1/\mu_v$	$\lambda_v$	PB <sub>res_voz</sub>	PB <sub>total</sub>
4	0,0267	60	0,0015	5 %	3%
4	0,02	80	0,0011	5 %	3%
4	0,0160	100	0,000902	5 %	3%
4	0,0133	120	0,00075	5 %	3%
4	0,0114	140	0,00064	5 %	3%
4	0,01	160	0,00056	5 %	3%
4	0,0089	180	0,0005	5 %	3%

**Número de slots fijo vs. PB<sub>total</sub> variable**

## CASO 1

	Nº slots reserva	$\lambda_{v_r}$	$1/\mu_v$	$\lambda_v$	PB <sub>res_voz</sub>	PB <sub>total</sub>
Cálculo 1	2	0,025	60	0,0075	31 %	1 %
Cálculo 2	2	0,046	60	0,023	50 %	4 %

## CASO 2

	Nº slots reserva	$\lambda_{v_r}$	$1/\mu_v$	$\lambda_v$	PB <sub>res_voz</sub>	PB <sub>total</sub>
Cálculo 1	3	0,035	60	0,0078	22,5 %	1,5 %
Cálculo 2	3	0,05	60	0,018	50 %	4 %

## CASO 3

	Nº slots reserva	$\lambda_{v_r}$	$1/\mu_v$	$\lambda_v$	PB <sub>res_voz</sub>	PB <sub>total</sub>
Cálculo 1	4	0,0183	60	0,000366	2,041 %	0,8%
Cálculo 2	4	0,03	60	0,00255	7,5 %	5,8 %

**Número de slots variable vs. PB<sub>total</sub> fijo**

## CASO 1

	Nº slots reserva	$\lambda_{v_r}$	$1/\mu_v$	$\lambda_v$	PB <sub>res_voz</sub>	PB <sub>total</sub>
Cálculo 1	2	0,03	60	0,011	36,65 %	1,5 %
Cálculo 2	3	0,035	60	0,0078	22,5 %	1,5 %

## CASO 2

	Nº slots reserva	$\rho_{v_r}$	$1/\mu_v$	$\rho_v$	PB <sub>res_voz</sub>	PB <sub>total</sub>
Cálculo 1	2	0,03	60	0,011	36,65 %	1,5 %
Cálculo 2	4	0,0183	60	0,000366	2,041 %	1,5%

## CASO 3

	Nº slots reserva	$\rho_{v_r}$	$1/\mu_v$	$\rho_v$	PB <sub>res_voz</sub>	PB <sub>total</sub>
Cálculo 1	3	0,035	60	0,0078	22,5 %	1,5 %
Cálculo 2	4	0,0183	60	0,000366	2,041 %	1,5%

Número de slots variable vs. PB<sub>total</sub> variable

## CASO 1

	Nº slots reserva	$\rho_{v_r}$	$1/\mu_v$	$\rho_v$	PB <sub>res_voz</sub>	PB <sub>total</sub>
Cálculo 1	2	0,0016	60	0,0000075	0,45 %	0,01%
Cálculo 2	4	0,11	60	0,055	50 %	41 %

## CASO 2

	Nº slots reserva	$\rho_{v_r}$	$1/\mu_v$	$\rho_v$	PB <sub>res_voz</sub>	PB <sub>total</sub>
Cálculo 1	2	0,15	60	0,12	80 %	30 %
Cálculo 2	4	0,0183	60	0,000366	2,041 %	0,8%